SOFTWARE SUPPORT FOR EXPERIENCE SAMPLING

A Thesis Submitted to the College of

Graduate Studies and Research

In Partial Fulfillment of the Requirements

For the Degree of Master of Science

In the Department of Computer Science

University of Saskatchewan

Saskatoon, CANADA

By

Mike Lippold

PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying, publication, or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other use of material in this thesis in whole or part should be addressed to:

Head of the Department of Computer Science
176 Thorvaldson Building
110 Science Place
University of Saskatchewan
Saskatoon, SK, S7N 5C9
Canada

ABSTRACT

User interface design is becoming more reliant on user emotional states to improve usability, adapt to the user's state, and allow greater expressiveness. Historically, usability has relied on performance metrics for evaluation, but user experience, with an emphasis on aesthetics and emotions, has become recognized as important for improving user interfaces. Research is ongoing into systems that automatically adapt to users' states such as expertise or physical impairments and emotions are the next frontier for adaptive user interfaces. Improving the emotional expressiveness of computers adds a missing element that exists in human face-to-face interactions. The first step of incorporating users' emotions into usability evaluation, adaptive interfaces, and expressive interfaces is to sense and gather the users' emotional responses. Affective computing research has used predictive modeling to determine user emotional states, but studies are usually performed in controlled laboratory settings and lack realism. Field studies can be conducted to improve realism, but there are a number of logistical challenges with field studies: user activity data is difficult to gather, emotional state ground truth is difficult to collect, and relating the two is difficult. In this thesis, we describe a software solution that addresses the logistical issues of conducting affective computing field studies and we also describe an evaluation of the software using a field study. Based on the results of our study, we found that a software solution can reduce the logistical issues of conducting an affective computing field study and we provide some suggestions for future affective computing field studies.

ACKNOWLEDGEMENTS

Many thanks go to Regan Mandryk and Carl Gutwin. I am constantly amazed at my good fortune for them accepting me as a student. They are both truly world-class researchers and mentors and I will forever be grateful for their guidance and support throughout my time in the Interaction Lab. The Interaction Lab members also deserve acknowledgement for sharing their experiences, knowledge and friendship, and many great discussions. Many of my experiences and friendships throughout the years have contributed to this work, and there are more people to thank than I can name – thank you all.

This thesis is dedicated to:

My brother Rick whose courage and spirit are an example to us all,

My wife Chi who has fulfilled my life in ways I could never imagine, and

My son Jacob who has brought boundless joy to my life.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS AND ACRONYMS

ARFF – Attribute-Relation File Format

CAS – Central Authentication Service

CSV – Comma Separated Values

DOM – Domain Object Model

DRM – Day Reconstruction Method

ESM – Experience Sampling Method

ESQ – Experience Sampling Questionnaire

FACS – Facial Action Coding System

GUID – Global Unique Identifier

GWT – Google Web Toolkit

IAPS – International Affective Picture System

JS – JavaScript

JSON – JavaScript Object Notation

PAWS – Personalized Access to Web Services

PCA – Principal Components Analysis

PDA – Personal Digital Assistant

RAM – Random Access Memory

SAM – Self-Assessment Manikin

SMOTE – Synthetic Minority Oversampling Technique

SOAP – Simple Object Access Protocol

UX – User Experience

XML – Extensible Markup Language

1 INTRODUCTION

User emotional states are becoming important in interface design. User experience is becoming an increasingly important aspect of interactive systems and can make the difference between a product that flourishes, and one that is discontinued. Traditionally, our understanding of user experience has been related to usability and productivity – considering measures such as the time taken to complete a task or the number of errors made. Recently, however, user experience research has incorporated measures related to aesthetics and emotion, focusing on the hedonics of interaction as opposed to the earlier pragmatic approaches [34]. Understanding the emotional responses of users to their interactive technology is quickly becoming a standard component of user experience testing.

Being able to determine emotional states can be used in three ways: to improve usability, to adapt to the user's state, and to allow greater expressiveness. As mentioned above, usability can be improved with a user experience perspective in which software is tested and users' emotional responses are measured; emotional responses can be used to inform software developers about areas of the user interface that require improvement. Another way to improve software is for it to adapt to a users' experience. For example, if a user is frustrated performing a task, an intelligent help agent could assist them. Third, hedonic user experience dictates that users should be able to naturally express emotions using their computer. People who use computers for social activities miss the essential emotional component used in face-to-face human interactions. There are replacements, such as emoticons, but they lack the expressiveness of real emotionally-based interactions. A computer that detects user emotional states can be used to evaluate users' experience and also to help improve the expressiveness of interactions. Unfortunately, the standard desktop computer is not currently able to detect emotion.

Making use of emotional states requires predictive modeling to recognize emotional states, which has several requirements. Studies in affective computing, "computing that relates to,

arises from, or deliberately influences emotions" [66], use an approach analogous to how people recognize emotion. When humans assess the emotional state of others, they begin by observing them: they may watch their face, listen to their voice, or notice their body language. Upon awareness of one or more of these signs, they match an emotion to the person that fits with the context of the situation and their previous experience. In affective computing, the observing phase involves capturing some kind of data such as a picture of a face. The match phase determines a person's emotion based on a machine learning construct called a predictive model. Predictive models learn to properly match by using examples that consist of emotion labels (e.g., level of anger) and some data about the user (e.g., a picture of the user's face). The model learns to recognize certain user data as the emotion label and once trained, can be used to recognize user data without emotion labels. For predictive models to learn to match well, they need many examples. It is important to note that in predicting the emotions of humans, neither humans nor predictive models are one hundred percent accurate.

Gathering emotional state data can be done in a laboratory, but lab studies lack realism. In laboratory settings, participants might be actors and asked to feel a particular emotion. Other studies might induce participants to feel an emotion by using pictures or video clips. It has been suggested that traces of emotions in user data may be weak or absent in laboratory studies because these studies lack realism [56]. Furthermore, laboratory studies are not scalable – only a small number of people can be tested this way. As a result of these limitations, we focus on field studies in this research.

Field studies are much more realistic, and can make use of real activity data such as the application being used, mouse and keyboard events, or other traces of interaction with a computer system. Field studies are more realistic because they are conducted in participants' natural environments. The experience sampling method (ESM) is a field study approach used in the social sciences for capturing "self-reports of mental processes" [18]. The ESM can be used to probe participants at random intervals to fill out questionnaires about their emotional states. Early ESM studies used pagers that notified participants to fill out questionnaires. More recent

studies have used mobile devices to gather ESM data. An affective-computing field study can use the ESM approach on users' desktop computers to capture the emotion labels.

However, it is logistically difficult to carry out field studies for emotional state modeling: it is difficult to gather activity data, it is difficult to collect ground truth data, and it is difficult to connect the two together. This leads us to our research problem.

1.1 Problem

The central problem addressed in this thesis is:

> ***It is logistically difficult to carry out affective computing field studies because it is difficult to gather user activity data, it is difficult to collect ground truth data, and it is difficult to connect the two together for analysis.***

More specifically, there are five roadblocks when conducting affective computing field studies:

1. **Data collection**. Gathering data from a small number of participants' computers (e.g. two or three) is relatively easy. However, as the number of participants grows, it becomes more difficult to reliably retrieve these data. Email servers can reject messages or become overloaded from large data files (e.g., 5MB) and participants may not comply with instructions to send their data files to researchers.
2. **Ground truth**. Measuring ground truth – the actual emotional state a participant is experiencing – is difficult. Laboratory studies can use triangulation, i.e., emotion can be determined by asking the participant and by measuring physiological signals that are known to indicate certain emotions. In a field study, physiological devices are less practical: although they could be deployed and participants could be instructed in proper usage of the devices, this approach is logistically difficult, intrusive, and expensive.
3. **Predictive modeling**. Data from participants must be in a format that can be used for predictive modeling. User data and ground truth must be matched together correctly.
4. **Uncontrolled factors**. In field studies, some of the factors influencing the emotional state of participants are uncontrolled. For example, a participant may feel frustrated for a

variety of reasons: because of the software they are using, because of their neighbours' loud music, or because their boss has not given them a raise, or a combination of these.

5. **Unknown frequency and duration of emotions**. Similarly, the frequency and duration of an emotional state is uncontrolled. For example, in a laboratory study we can know whether a participant is angry or tired by inducing these states. We can use established experimental procedures to help ensure that our statistics will be valid. We can have fifty trials with each participant in a frustrated state and another fifty in a non-frustrated state. With a field study, we do not know how many times a participant feels a specific emotion and how long that emotional state lasts.

1.2 Solution

Our solution to the research problem is to address the first three roadblocks by:

> ***Creating a software system that supports field methods for affective research, including processes of gathering user activity data, collecting ground truth data, and connecting these datasets.***

1.3 Steps in Our Solution

We realize that other problems have similar technical requirements and learning about those problems and leveraging their solutions will help us solve our problem. Our solution combines other existing solutions in a new way. The following steps were used to implement and evaluate our solution:

1) *Survey existing solutions to similar problems*. The ESM was examined in other contexts. Techniques for gathering data from remote locations were investigated. Approaches to predictive modeling were also examined.
2) *Create an approach for affective computing field studies using the ESM*. The process of gathering data through to predictive modeling was examined and an approach suitable for affective computing field studies was developed.

3) *Design and implement support software*. We created an architecture and design for capturing participant data, transforming the data, and performing predictive modeling. Based on this architecture we built software to perform an affective computing field study.
4) *Evaluated the approach and software by conducting a field study.* We gathered affective field data from 26 participants over eight weeks to test the approach and the software.

1.4 Evaluation

The software support developed in this thesis was evaluated using a field study. There were two main goals of the study:

1) *Validate our approach for detecting participants' emotional states*. The central questions we asked were:
    - Can we determine ground truth?
    - Can we collect data for predictive modeling?
    - Can we build predictive models from the data?
    - Can we reduce the logistical difficulty in carrying out these processes?
2) *Determine whether any emotional states can be detected in participants.* The purpose of our approach was to enable affective computing field studies, so the question was:
    - Can we detect emotional states using our approach?

1.5 Contributions

The main contributions we provide are:

1. We designed, implemented, and successfully tested a support system for conducting experience sampling studies for affective computing.
2. We identified several recommendations for future experiments based on our initial study.

1.6 Thesis Outline

The remainder of this thesis is organized as follows:

- Chapter 2 presents a survey of related research, which forms the basis for the research presented in this thesis. We define emotion, mood, affect, and emotional state. Two theories of emotion are presented. Techniques for determining emotional states are described and three methods of establishing ground truth emotional states in studies are presented. The ESM and predictive modeling are described and some evidence is given for the plausibility of detecting emotional states using mouse dynamics.

- Chapter 3 describes the software we created for collecting keyboard, mouse and questionnaire data. The requirements for recording, aggregating, and downloading data are described as well as requirements that arose while conducting the field study. We describe the overall system architecture. The design details of the client logging software, server-side web service, data retrieval web application, and daily reporting scripts are also all described in detail.

- Chapter 4 describes our field study. We explain the selection of our participants, and the hardware and software requirements we imposed. The procedure for collecting the mouse and questionnaire data is described as well as the procedure for summarizing the mouse data into analyzable metrics. The method employed for the statistical analysis is described.

- Chapter 5 reports the results of our field study.

- Chapter 6 discusses the software and the results of the field study. Our main findings are summarized.

- Chapter 7 summarizes the research presented in this thesis. Our contributions are described and we discuss some future directions for the continuing study of detecting user affect from mouse data.

2 RELATED WORK

In this chapter, we provide an overview of previous works that are relevant to this thesis. We begin by addressing the terms emotion, mood, affect, and emotional state. We describe the two main emotional models, discrete and dimensional. Third, techniques for determining emotional states are discussed. Next, we describe how studies control for the emotional state (ground truth) that participants experience. We describe the relationship between computer science and the ESM. The classification of emotional states using predictive modeling is described. Finally, we present some studies that have examined mouse usage and user characteristics.

2.1 Emotion, Mood, Affect, and Emotional State

Three key concepts are discussed in the study of emotion: emotion, mood, and affect. There are many definitions of emotion [47]. Some focus on the cognitive experience, some the physical experience, and others combine the two. We consider emotions and moods to describe the cognitive aspect, affect to describe the physical aspect, and emotional state to describe the combined cognitive and physical state of experience.

Emotions and moods differ in that emotions arise quickly and are short-lived [23], whereas moods develop slowly and are longer-lived [66]. Examples of emotions are anger, enjoyment, and fear. Moods are generally described as positive or negative, e.g., being in a good mood.

Affect is the physiological (physical) manifestation or external display of an emotion or mood [1] and is an important concept because if emotions have a physical representation, then it may be possible to measure it. And indeed, physical characteristics of emotion have been studied for many years, perhaps most notably using facial expressions [7,16,20,22,26,65], but other modalities as well such as skin conductance, heart rate, and muscle activation [21,57].

2.2 Discrete versus Dimensional Models of Emotion

Theories of emotion have an impact on how emotion information is gathered, and two main views have arisen in the theory of emotions. Discrete emotion theories describe emotions as individual, separable experiences [22,76]. Dimensional models have also been proposed ([71] as described by [69]), one of the more popular being the Circumplex Model [69]. In this section, we go into more detail on both of these.

2.2.1 Discrete Emotion

The central idea of discrete emotion theories is that emotions are separate experiences, they do not overlap, they can be individually measured, and they are targets of evolution. Tomkins [76] describes affect as amplifying drive and being selected for or against in an evolutionary sense. For example, when one feels suffocated, fear increases the drive to breathe. Emotions, thus, can have a biological role of increasing or decreasing the chance of survival of an individual.

Tomkins refers to the face as the "primary site of affects" and Ekman [22,24] also focuses on the face and facial expressions as the main indicator of the affect system. The Facial Action Coding System (FACS) [24] identifies discrete, cross-cultural emotions using sequences of facial muscle activations. We discuss this more later.

Considering emotions as discrete means they are identified through labels such as fear and enjoyment, which can be observed internally by individuals and gathered using self-report, or externally, for example using facial expressions.

2.2.2 Dimensional Models of Emotion

Dimensional models of emotion map emotions on some multi-dimensional space. A widely used model, the Circumplex Model [69], uses two dimensions: arousal (activation-deactivation) and valence (pleasure-unpleasant). Figure 2.1 shows the Circumplex Model with the y-axis displaying activation/arousal and the x-axis depicting pleasantness/valence. In this diagram, fifteen emotions are labeled in a circle indicating their levels of arousal and valence, which were

determined in experiments in which participants were asked to sort various emotional states in a circular order onto the two-dimensional chart [69].

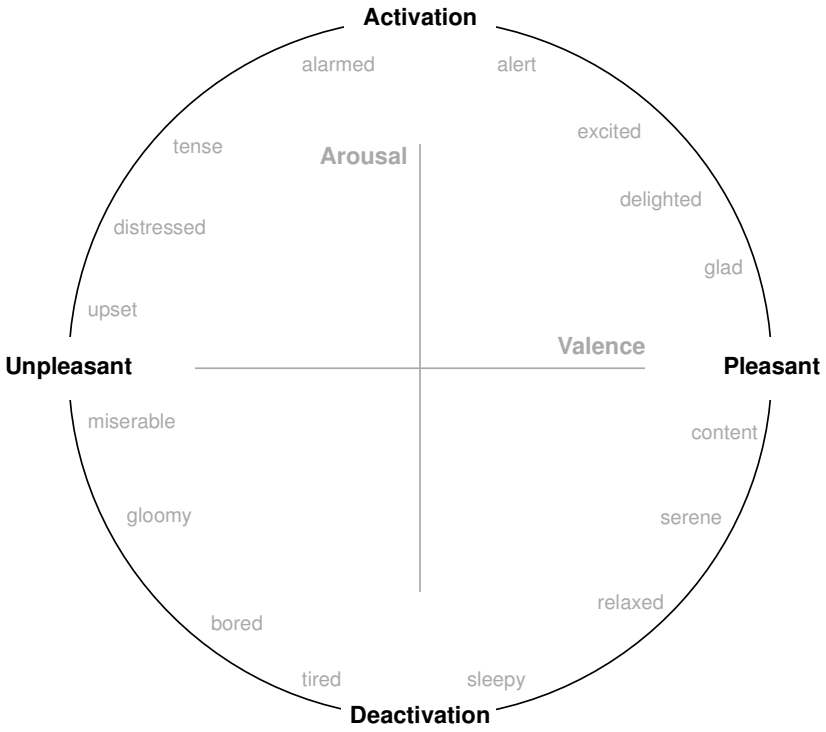


Figure 2.1 - The two-dimensional circumplex emotional model based on Russell [69] (used with permission from [60]).

Strictly speaking, discrete emotions can be considered a multi-dimensional model where the number of dimensions is the same as the number of emotions.

2.3 Determining Emotional States

In this section, we discuss three ways of determining the emotional state another individual is experiencing: self report, facial expressions, and physiological signals.

2.3.1 **Self Report**

Self report is the simplest form of determining a participant's emotional state – participants tell a researcher how they are feeling. Perhaps the most basic reporting of emotions can be done using a diary. However, this data is not structured and can be difficult to analyze and interpret. More structured techniques using questionnaires ease this effort. Another consideration is the timing of the emotional state. When the emotion is experienced relative to when a participant is asked about their emotional state is important because recalling experiences from an earlier time is often error prone. Better results can be obtained by asking participants at the time researchers want to know their emotional state.

The theory of emotion used by researchers impacts how they will record emotional states. Discrete emotions can be determined with adjectives describing emotional state levels using a questionnaire and a Likert scale [55]. The arousal-valence dimensions can be measured in a few ways such as using a semantic differential questionnaire of arousal and valence, a semantic differential scale of adjectives categorized by arousal and valence [59] (see Figure 2.2), and non-verbal, pictorial representations such as the Self-Assessment Manikin (SAM) [11]. Hybrid approaches also exist that take into account both discrete and dimensional models. For example, Tellegen et al. provide a hierarchical model that takes into account valence at the top-level, arousal in the middle, and discrete emotions at the most detailed level [74]. In a study to correlate mouse motions with emotional states, Zimmerman et al. used the Self-Assessment Manikin to determine participants' emotional states after showing them evocative pictures [82].

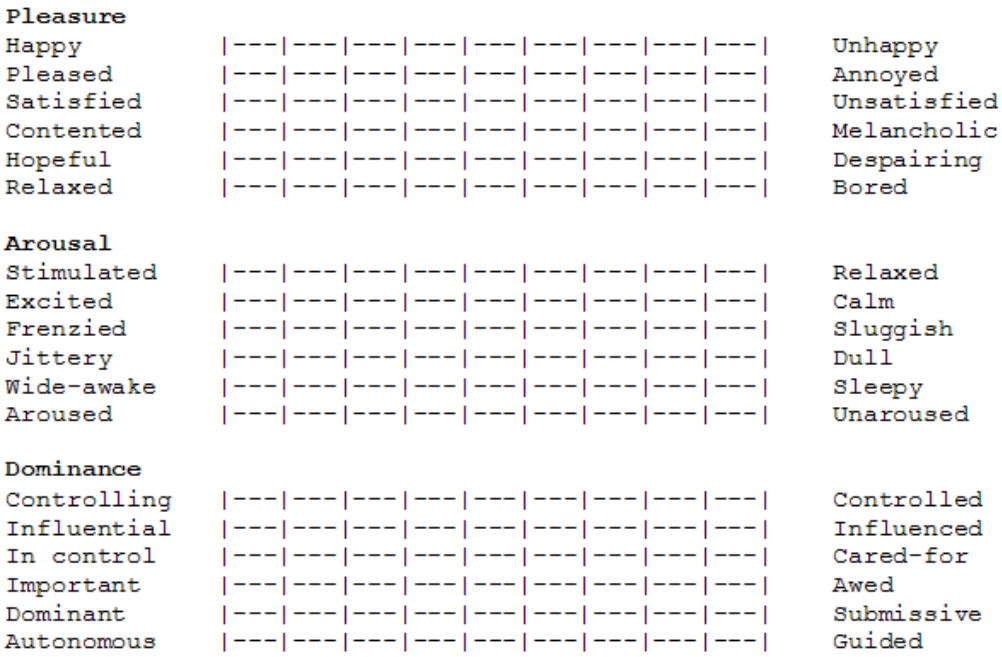


Figure 2.2 - Semantic differential questionnaire for emotional states. Each scale has a rating from 4 to -4 and results are summed in each section (Pleasure, Arousal, Dominance) to give a dimensional score. (based on Mehrabian and Russell [59])

2.3.2 Facial Expressions

People use facial expressions to communicate and read others' intents and emotions. There is evidence that some facial expressions are cross-cultural [22] and a system, the FACS (Facial Action Coding System), has been devised for observers to identify these emotions in subjects [24]. This system identifies distinct facial muscle activations, called action units that, in combination, help identify emotional states. In the FACS, a smile is considered a combination of action units that indicate happiness which may represent genuine happiness in a subject, but could also be a social grace or have a deceptive intent. Facial expressions can be inexpensively recorded using video cameras; however, the possible deception involved in facial expressions makes it problematic when trying to ascertain how participants are really feeling. Facial expression recognition has been studied in computer science, for example Essa and Pentland

used videos of faces to identify FACS action units and a motion energy representation which were used in combination to determine facial expressions [26].

2.3.3 Physiological Signals

As well as having a physical manifestation through facial expressions, emotions are also evident through physiological manifestations of the nervous system [49]. Heart rate, hand temperature, skin conductance, muscle tension and brain activity have all been shown to differentiate some emotions from others and/or help distinguish between high and low arousal and between positive and negative valence [21]. Affective computing studies have also successfully used physiological measures to distinguish between emotional states [31,42,56,70]. A drawback of measuring physiological signals is that specialized equipment is required, which is often expensive and intrusive, but compared to facial expressions, physiological signs are more difficult to fake.

2.4 Establishing Ground Truth in Studies

To conduct studies of emotional state, researchers need to know the emotional state experienced by participants – they need to know the ground truth emotional state of participants. In this section, we describe some of the common techniques for establishing ground truth emotional states of participants in research studies.

2.4.1 Method Actors

The facial expressions and physiological responses of method actors have been used in emotion studies [66]. Using this technique, researchers ask actors to feel a certain emotion, remember certain situations that evoke an emotion, or even make certain facial expressions to get them to feel a particular emotion. It is a very convenient technique, allowing researchers to test many emotions at one sitting and while this has allowed research to make progress, several researchers note the lack of realism of actors compared to individuals in realistic situations [7][9][19].

2.4.2 **Emotional Elicitation / Mood Induction**

One way to control the emotions of experiment participants is to use stimulus to induce a particular emotion or mood. Two common techniques for invoking emotion are showing pictures or movie clips. The IAPS (International Affective Picture System) has been created which is a library of pictures that have known emotion inducing effects [10]. The IAPS images have been used in many studies, validating the invoked mood across many participants. Movie clips have also been used as an alternative to induce emotional states [68]. The same authors of IAPS have other libraries of media for inducing emotional states, including sound [12], words [13], and text [14]. Sometimes, multiple stimuli can be used in concert to try to strengthen the induced emotional effect. For example, sadness was invoked in a study of emotional contagion in instant messenger use by having participants watch a clip from Sophie's Choice, solve difficult anagrams, and listen to sad music [33]. An alternative to displaying a stimulus is to invoke a mood by explicitly manipulating the experimental environment. For example, in an affective computing experiment, frustration was invoked by freezing the participants' mouse cursor, making the mouse unresponsive, similar to a slow computer or a system error [67].

In many cases emotional elicitation works in the laboratory, but the quality and intensity of participants' emotional states could be quite different in more realistic environments. Furthermore, there is a limit to the number of emotional states participants can be induced into, and this limits the number of emotional states that can be studied at one sitting.

2.4.3 **The ESM (Experience Sampling Method)**

The ESM (Experience Sampling Method) involves asking participants to provide information about their experience at pre-determined times such as at fixed intervals during the day (interval-contingent), after certain events have occurred (event-contingent), or when signaled (signal-contingent) [78]. For example, a participant could be instructed to carry a pager and blank sheets of questionnaires to answer. When the pager beeps, the participant would fill out the questionnaire. Signal-contingent reporting is the most often reported technique in the literature and in this thesis, we mean ESM with signal-contingent reporting.

The ESM can provide ecological validity that other techniques are unable to provide [18]. The emotions experienced by participants are real, not induced by pictures, movie clips, or other types of manipulation. The accuracy of ESM reports is greater than self-report studies in which participants answer questionnaires long after they have experienced the states they are asked about [61]. In fact, the development of experience sampling was partly motivated by dissatisfaction with self-report experiments where participants could not accurately recall past experiences [18].

There are drawbacks to using the ESM. First, while the ESM has been reported as more accurate than self-report, it is dependent on participants' willingness (motivation) to provide accurate information [18,61] and this may select for certain types of individuals who are willing to participate and who will finish a study that uses the ESM [61]. However, this problem exists to some extent with other methodologies as well. Whether in a laboratory (where experimenters are present) or in participants' homes or work places (where researchers are absent), participants require some level of motivation to participate in any study.

Second, the states experienced by participants are out of the control of researchers. A participant may never feel angry during an ESM study whereas in a controlled laboratory setting, a balanced design can help ensure each state occurs the same number of times.

2.5 The ESM in Computer Science

The ESM and computer science complement each other and in this section, we report how the ESM has benefited from technology and how technology is benefiting from ESM.

2.5.1 Computerized ESM

Modern computing devices can greatly assist studies using the ESM. ESM relies on signaling at intervals decided upon by researchers. While pagers were used in early ESM studies, technology has advanced considerably. Mobile devices can be leveraged to assist with signaling and capturing participant data for ESM studies. Many people now carry some form of computerized mobile device such as mobile phones, smart phones, and personal digital assistants (PDAs).

These can be used as ESM signaling and recording devices, both by notifying participants when it is time to fill out a questionnaire and by providing the questionnaire to be filled out [29]. Because many devices are connected to a cellular or wireless network, data can be sent to a data collection server. Still, a generalized and extensible desktop system for collecting customizable ESM data and user interaction data (e.g., typing, mouse use, window focus) does not yet exist.

2.5.2 UX (User Experience)

UX (User Experience) is a trend in the engineering of technology that focuses on non-functional aspects that encapsulate the emotional and aesthetic experience of using technology [52]. In the UX community, there is a recognized need to evaluate products using field studies to gauge how real systems perform in real life [48,64] and a number of UX studies have used the ESM [29,39].

There are drawbacks of using ESM in UX. It is rigid in that it usually involves questionnaires, so qualitative feedback from participants such as commentaries are not captured. One recent UX study used the Day Reconstruction Method (DRM) – daily written summaries of participants' use of a product during the day [43]. The researchers chose DRM over ESM because they wanted to capture more detail about participants' experience. And more generally, programming and setting up computerized ESM is more effort than using more primitive methods such as a pager and paper questionnaires.

2.6 Predictive Modeling

In this section, we will describe the main concepts of predictive modeling to give the unfamiliar reader a better understanding.

Predictive modeling is important because it is the main technique used in affective computing to determine emotional states from user data. In predictive modeling, the goal is to provide a model that can distinguish between two or more *classes* [79]. An example of this is classifying level of enjoyment of a person into high and low enjoyment. To distinguish between these two classes of enjoyment, some data is required that will allow us to differentiate between classes. If our data is a picture of the facial expression of an individual, we might use the existence of a smile to

distinguish between someone who is experiencing a high level of enjoyment and someone who is experiencing a low level of enjoyment. In the language of predictive modeling, the data that indicates the existence or not of a smile is called a *feature*. In this case, it might just be a true or false value; however, a smile could also be described by a continuous or ordinal variable based on the degree of smile. A *model* can be created that distinguishes between classes by using the smile feature. If a smile exists, the person is classified as having high enjoyment; otherwise, the person is classified as having low enjoyment. In more realistic predictive modeling cases, there is usually more than one feature. Notice in our example that to perform the classification described we need to know the actual class for the data. The combination of the actual class and the set of features is called an *instance* and it is common when building predictive models that many instances are required. Of course, the end-goal of a predictive model is to classify instances without knowing the actual class. The actual class is used to train and evaluate the model.

Another part of predictive modeling involves the use of *machine learning algorithms* to determine the best model for classification. In the above example, we describe a decision-tree algorithm, but there are various types of algorithms that can be used. The common element of the process, regardless of algorithm, is the generation of a model that has the highest *prediction rate*. The process also involves what is called a *training stage* and a *testing stage*. During the training stage, the algorithm uses *training data* of many instances to determine the model with the best prediction rate. During the testing stage, *testing data* is used to evaluate how well the model performs. By doing this, the generalizability of the model is tested. This is important because one of the challenges of predictive modeling is ensuring that models do not *overfit* training data [79]. Overfit occurs when a model is trained too specifically for the training data and does not generalize well to other data.

Another important part of the predictive modeling process is *feature extraction*. In the example we gave above, a smile was used as a feature, but we did not indicate how the smile was determined in the pictures. We would perform some kind of image processing to determine

whether a smile exists or not. The process of generating features from raw data is called feature extraction.

A number of software solutions exist that automate the process of predictive modeling; MATLAB and Weka are most notable.

2.7 Classifying Emotional States Using Mouse Dynamics

The computer mouse is a ubiquitous device on desktop computers that can be used to classify emotional states. As we have already indicated, there are correlations between physiology and emotions. Heart rate, hand temperature, skin conductance, muscle tension and brain activity, among other physiological measures are known to change when people are in different emotional states. So, there is a physiological basis for emotions. In this section, we discuss research that indicates it is plausible to classify emotional states from the mouse.

Studies have found a correlation between mental stress and muscle activation. Handwriting has been found to be differentiable based on whether a person is stressed or not [45]. Mental stress has been found to correlate to increased blood flow and muscle tension in the shoulder muscles [50]. Muscle activation of shoulder, neck, and forearm muscles has been found to be greater when participants are performing tasks with greater mental demands [51]. The link between greater or lesser muscle activation and motor activities is not unreasonable, so at least for stress it is plausible that there may be differences between how participants move or click their mouse. And this has indeed been found to be the case. The pressure applied to a mouse has been shown to vary with valence [46] – the Circumplex Model (Figure 2.1) indicates stress is a negative valence emotional state. We believe the evidence for muscle changes correlated to stress and valence may also indicate that other muscle changes can occur for other emotional states and that these are measurable from mouse dynamics.

Mouse dynamics refers to moving and clicking the mouse. The metrics associated with mouse dynamics may include distance, velocity, acceleration, jerk, and speed of mouse click. Mouse dynamics may indicate emotional states because emotional states may affect motor function. Studies have been able to distinguish between participants who have different motor function

capabilities. Distinct differences have been found between mouse use of adults, seniors, and Parkinson's patients [38,44]. Expertise level of users has also been differentiated using predictive models on mouse movements, clicks, plus contextual information such as the number of menus opened [37,58].

2.8 Summary

The differences between emotion, mood, and affect have been explained. Emotion is a cognitive experience of feeling that is short in duration compared to moods. Affect is the physical feeling of emotion. We call an emotional state the combined cognitive and physical manifestation of emotions. Discrete and dimensional models of emotion were described. Discrete emotions are separate experiences that have labels such as anger whereas the two-dimensional Circumplex model describes emotions in terms of valence and arousal. Three methods of determining emotional states are described: self report, facial expressions, and physiological signals. Three techniques of establishing ground truth in experiments are also discussed. Method actors and emotional elicitation are convenient techniques that can be used in the laboratory, but lack realism. The ESM (experience sampling method) provides ecological validity that method actors and emotional elicitation cannot, but ground truth is unpredictable and depends on participants' experiences during a study. Computer science and the ESM are complementary and have been used together to create computerized ESM implementations as well as in UX (user experience) studies. Predictive modeling allows emotional states to be classified by computer systems, but requires instance data be recorded with ground truth classes. Mouse dynamics may be a promising modality for classifying emotion because some emotional states and muscle activation are correlated, and studies have detected differences between participants with different physical abilities using mouse dynamics.

3 FIELD STUDY SOFTWARE

In order to capture mouse data from users' computers, we created specialized software that allowed us to record and collect mouse activity data[1]. Our software also provided a way for users to contextualize their data using subjective labels of emotional state. We also created software for retrieving and summarizing our data for analysis using statistics and predictive modeling.

In this chapter, we describe the software system we created for collecting ESM data, retrieving the data from a central server, and summarizing it for later analysis, specifically predictive modeling. Further detail is provided in Appendix C. We start by describing the requirements and follow with a detailed description of each part of the entire system we developed.

3.1 Requirements

At a high-level, our system facilitated the process of collecting, summarizing, and analyzing data – a process common to many experiments (Figure 3.1). The software we created fits into the data collection and summarization categories, as off-the-shelf software (Excel, SPSS and Weka) was used for the analysis. The rest of this section describes the requirements for data collection and summarization.

[1] The software also collected keystroke activity which was used as part of another study, but in this thesis, we focus on recording and analyzing mouse data only.

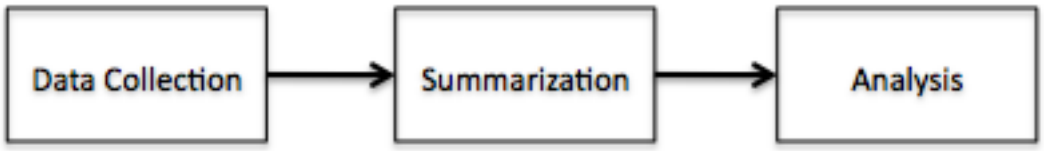


Figure 3.1 - Experiment process used for field study

3.1.1 Data Collection Requirements

This section outlines requirements for the data collection software.

3.1.1.1 Record mouse cursor coordinates and mouse button up and down events from all applications

An important function of the system was to capture mouse activity from users. This requirement included the recording of mouse cursor coordinates when the mouse was moved and when mouse clicks occurred. The requirement also included the recording of mouse button up and down events. A timestamp was included for all events that were recorded and logged.

3.1.1.2 Provide an emotional state questionnaire for users to fill out

The ground truth of the emotional state of users needed to be established. The simplest way to accomplish this was to ask them [55]. We chose to use subjective emotional state labels (e.g., frustrated, angry, etc.) to determine the ground truth emotional state of participants because other techniques, such as measuring physiological signals, are costly, intrusive, and require logistic support, such as training participants and replacing bad sensors. Asking participants about their emotional state was inexpensive, unobtrusive, logistically simple, and has been shown to be a reliable indicator of participants' emotional states [18].

We devised a questionnaire in which we asked users to rate how they felt, using the fifteen statements in Table 3.1.

Table 3.1 - Statements used in questionnaire to determine users' affective state

I am frustrated	I feel relaxed
I am focused	I feel excited
I am angry	I am distracted
I am happy	I feel bored
I feel overwhelmed	I feel sad
I feel confident	I feel nervous
I feel hesitant	I feel tired
I feel stressed	

As recommended by Lindquist and Barrett [55] for gathering discrete emotion data, each statement was answered on a five-point Likert scale: strongly disagree, disagree, neither agree nor disagree, agree, and strongly agree. Users were asked to fill out the questionnaire every hour, but only if they were actively using their computer. We defined a participant as actively using their computer if there were more than 2,000 mouse and/or keyboard events over a five-minute period. This means that if mouse motion events were captured every 10 milliseconds, the participant was moving their mouse continuously for 20 seconds. We believe this is suitable for capturing mouse data because we observed that mouse motion tends to occur in bursts rather than continuously. We also observed during the test and pilot phases that a higher value, 3,000 events, resulted in the questionnaire appearing infrequently, sometimes not at all. However, because the 2,000 events is a combination of mouse and keyboard, it was possible to receive a larger number of keyboard events and fewer mouse events or vice versa and still meet the 2,000 event threshold.

3.1.1.3 Provide a demographic questionnaire for users to fill out

When users started the recording software for the first time, we required them to fill out a one-time demographic questionnaire before they started to use the software. We collected standard demographic information (e.g., age, sex, and occupation) and some information specific to our study (e.g., dominant mouse hand, computer expertise). As described in the next chapter, we used a web site that allowed participants to check a consent form and provide their email address. However, we wanted the ability to correlate demographic information with the questionnaire and mouse data collected. To accomplish this and preserve participants' confidentiality, we captured demographic information the same way we captured participants' mouse data and subjective emotional state labels – using the software.

3.1.1.4 Users should be able to disable the software

While we wanted to capture as much data as possible in as many different situations as possible, we realized it was important to respect users' privacy and to minimize the software's disruption to them by allowing them to disable the software, particularly logging of keystroke data, at any time.

3.1.1.5 Users should be able to skip the emotional state questionnaire

Again, to respect users' privacy and minimize disruptions, we thought it was important to allow users to skip the emotional state questionnaire.

3.1.1.6 No noticeable impact to the system or other applications

It was important that the logging software should not impact mouse cursor movement, system performance or the performance of other running applications. If there was a noticeable impact on performance, we may have influenced the emotional state of users and lose the benefit of unobtrusively recording users' behaviour.

3.1.1.7 Mouse data should be captured at the highest rate possible

This was important because if users in a particular state had a tendency for small jittery movements (as described in Chapter 2), we wanted to capture the associated data. Recording data at the highest data rate would provide the greatest chance of capturing small changes in mouse movement behaviour.

3.1.1.8 Data should be reliably delivered to a central server

We recognized early in our development process that the most reliable way to collect data was to deliver it automatically from users' computers to us. Considering the large data files, which were up to 4MB per participant for a day of data, we knew that asking users to email large data files to us may have a problem for mail servers. Delivering data automatically through the collection software would avoid overloading mail servers, reduce unnecessary effort from users, ensure the data was received in a consistent format and prevent receiving redundant data. Because the research data collected was shared between two projects, one modeling mouse dynamics and another keystroke dynamics, it was valuable to make the data available to multiple researchers.

3.1.1.9 Researchers must be able to retrieve data from the server in a secure manner

After the data was delivered to the server, we needed to retrieve it in a secure fashion. As a matter of practicality, the researcher should be able to easily download all collected user data for a study and differentiate users' data from each other.

3.1.1.10 The system should be configurable to stop collecting data

When the study was over, we wanted the ability to ignore new data sent from the same user. This was important for maintaining the integrity of the data after the study was finished.

3.1.1.11 A daily report should indicate whether users answered the demographic questionnaire

After the field study began, we realized that additional functionality would make the management of the field study easier. A daily report would show participants who failed to complete the demographic questionnaire. After a few days, we could investigate to ensure the software was working and that users did not want to opt out of the study.

3.1.1.12 A daily report should indicate the number of questionnaires answered by users

We were concerned that participant computers might have issues sending data to the server, so we wanted to keep track of the daily progress of users who filled out the questionnaires. This also provided an indication of whether we needed to recruit more participants.

3.1.1.13 Participant anonymity should be preserved through the entire experiment process

A common requirement for many experiments is to preserve the anonymity of participants. The overall approach including the software system should not allow researchers to determine the identity of participants.

3.1.2 Summarizing Data for Analysis

The raw mouse and questionnaire data received from users' computers were impossible to analyze without some manipulation. For the purposes of analyzing the collected data, we decided to summarize the data for the five-minute period immediately before the questionnaire appeared on users' computers. This summarization did not need to happen until just before the analysis phase of the project, so we collected the raw data and performed the summary on the full data set. The mouse-click and mouse-motion summarizations we performed are shown in Table 3.2 and Table 3.3.

Table 3.2 - Mouse click summarizations

Left click count
Right click count
Total click count
Single click count
Single click dwell time* mean, standard deviation, median and maximum
Double click count
Double click – first click dwell time* mean, standard deviation, median and maximum
Double click – second click dwell time* mean, standard deviation, median and maximum
Double click – time between first and second click mean, standard deviation

* *Dwell time* is the time between mouse down and mouse up events

Table 3.3 - Mouse motion summarizations

Total distance
Speed – mean, standard deviation, median and maximum
Acceleration – mean, standard deviation, median and maximum
Jerk – mean, standard deviation, median and maximum
Still moment* count
Still moment* duration

* A *still moment* occurred when mouse activity ceased for 1 second or longer

3.2 Overall System Architecture

We implemented the first and second steps in the field study process – data collection and summarization – using five different custom software components (Figure 3.2). We envision this architecture to be durable and our implementation of it to be a proof-of-concept that the architecture is robust and solves our problem.

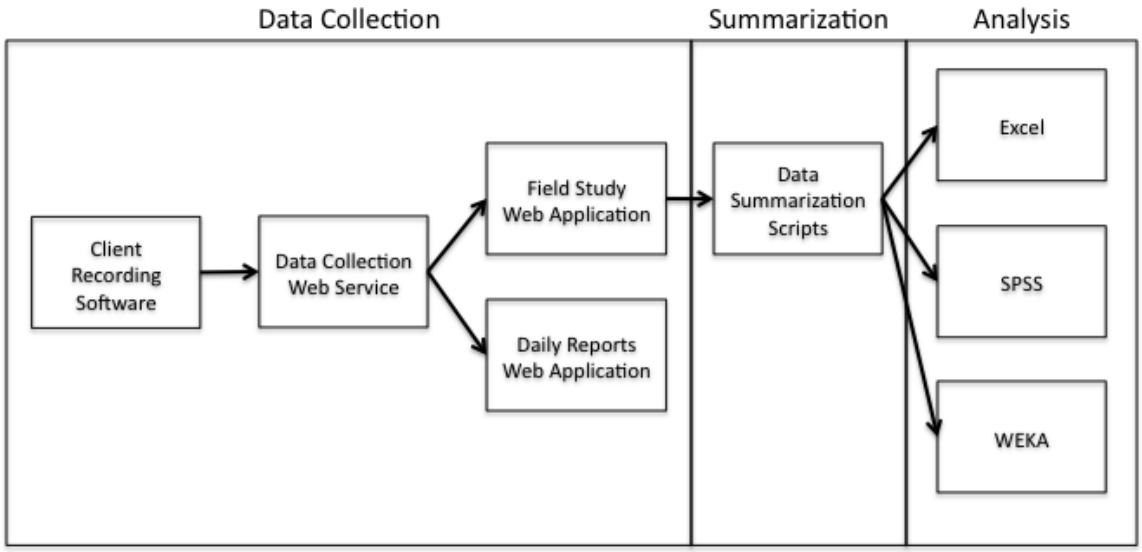


Figure 3.2 - Data flow through the various software components

The data collection process was implemented using four software applications.

1. The **Client Recording Software** was responsible for recording mouse activity, gathering questionnaire responses from participants, and sending these data to a data collection server.
2. The **Data Collection Web Service** primarily received data from the many instances of the Client Recording Software deployed on participants' computers.
3. The **Field Study Web Application** allowed researchers to view and download data received from participants' computers.
4. The **Daily Reports Web Application** was responsible for sending daily reports to researchers and allowed researchers to view the history of daily reports.

The summarization process was implemented in a single software application.

5. The **Data Summarization Scripts** were written in MatLab and aggregated the data collected from participants into a format suitable for statistical analysis and modeling.

The analysis process was carried out using a variety of software tools including Excel and SPSS for statistical analyses, and WEKA for predictive modeling.

3.3 Client Recording Software Design

The purpose of the client logging software was to perform part of the data collection function by recording mouse events, logging them, and delivering them to a central server. Based on separation of concerns, our design of the software is divided into three major tiers: probe, log, and deliver (see Figure 3.3).

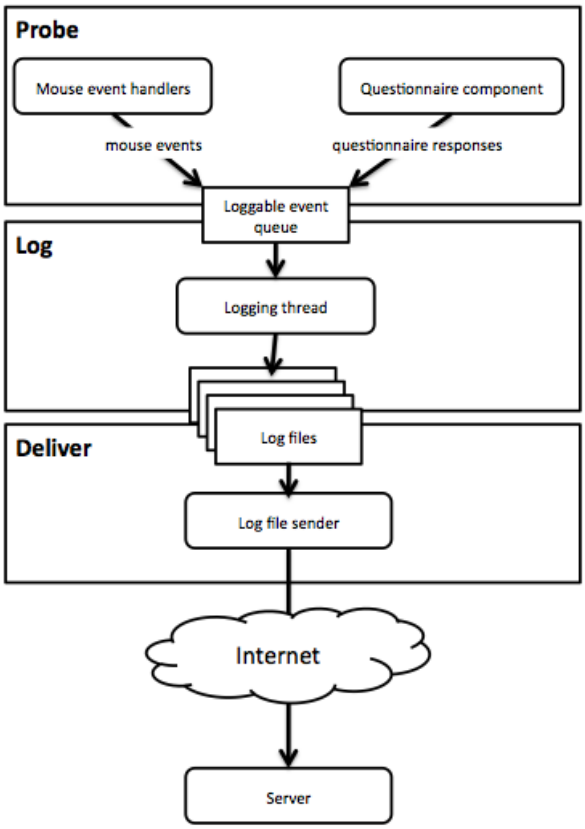


Figure 3.3 - Client logging software's three-tier design

On each user's computer, mouse activities and questionnaire answers were probed. Mouse motion events were received at varying rates, depending on participant mouse use. As mentioned in the previous section, questionnaire events occurred every hour, but may not have occurred if users were away from their computer, they chose not to answer the questionnaire, or they disabled the client software.

Since mouse activity, specifically mouse motion, occurred at a high rate (approximately every 8 milliseconds when users were continuously moving the mouse), it was important to minimize the impact from for the mouse event handling routines to the underlying operating system. We achieved this by storing all events to an in-memory log queue and allowing a separate logging mechanism to save the queued events to the hard drive. The logging mechanism consisted of the

log queue and a thread that periodically processed the events in the queue. Each event was saved to an appropriate log file for the event. For example, mouse motion events in the queue were saved to a mouse motion log file and questionnaire events were saved to a questionnaire file.

3.3.1 **Probe Tier**

From the software engineering perspective, two major event classes were specified based on the type of events: mouse activity and questionnaire. Each of the classes implements the ILoggableEvent interface. An ILoggableEvent object contains information to be logged to a log file. Specifically, the information was stored and accessed through the LogMessage property and the HeaderText property within an ILoggableEvent object. The LogMessage property generated text suitable for writing to a single line in a log file while the HeaderText generated text for the header line in a log file (Figure 3.4).

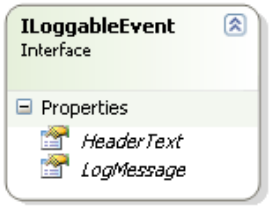


Figure 3.4 - ILoggableEvent interface definition

Each event type (e.g., QuestionnaireEvent and MouseMotionEvent) implemented the ILoggableEvent interface (Figure 3.5). An abstract type, AbstractLoggableEvent exposed two common properties used by all subtypes: ParticipantId and Timestamp. Another abstract type, AbstractMouseEvent, encapsulated common properties of all mouse subtypes. Each mouse subtype had private properties to log specific events such as the mouse button clicked (left or right) and a mouse up/down indicator.

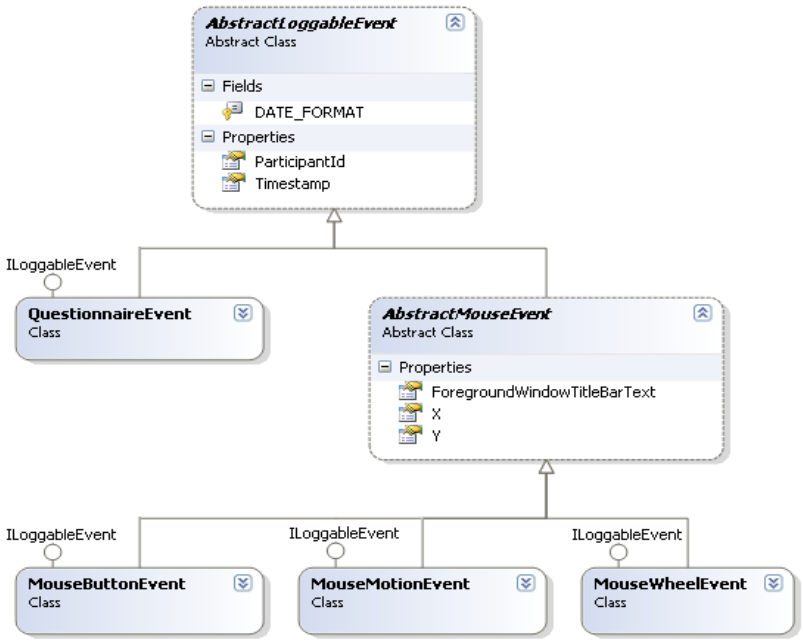


Figure 3.5 - Class hierarchy of some ILoggableEvent implementations

3.3.2 Log Tier

When the probe tier captured an event, it was responsible for sending a "Log" message to a LoggableEventWriter singleton. LoggableEventWriter managed a collection of queues for each event type and was responsible for adding new ILoggableEvents to the appropriate queue. When the emotional states questionnaire was answered, the LoggableEventWriter singleton wrote the contents of all queues to the appropriate log file. Mouse motion, mouse button, mouse scroll, and questionnaire events were all stored in separate log files.

3.3.3 Delivery Tier

In this tier, a DirectorySender class was responsible for sending log files. The web service received a single file at a time and stored it to a directory in the file system corresponding to the relevant experiment and participant. Web services use the HTTP protocol and HTTP was desirable because HTTP is most often used by web browsers and firewalls rarely block HTTP

traffic. A service created using other ports could be blocked, so we chose the common and rarely blocked HTTP protocol. We chose web services because it is an established standard and simplified development, however we could have used other standards such as JSON (JavaScript Object Notation) [17] and REST (Representational State Transfer) [27].

3.3.4 Client Software Implementation

A C# (.NET Framework 3.5) application was written for the Microsoft Windows XP and Vista operating systems to capture mouse and keyboard activity, and subjective emotional state labels. After the installation of the software, a demographic questionnaire was given, the software started and automatically launched after every computer restart. Table 3.4 outlines the data and frequency of recording.

Table 3.4 - Data captured by client software

Data Captured	Frequency of capturing data
Mouse pointer coordinates and button clicks	Continuously (events received about every 8 ms)
Owner process of open windows	Every 10 seconds
User state questionnaire	Every 60 minutes
System configuration parameters	After every questionnaire was completed

Low-level operating system hooks were used to continuously capture screen coordinates of the mouse pointer and all mouse button events. Low-level hooks allowed us to capture mouse events from all applications. Mouse cursor screen coordinates and button events, along with timestamps, were saved to a log queue that subsequently saved to a log file when the emotional states questionnaire was completed. The log queue buffer helped avoid I/O waits in the event thread and minimized the wait time while mouse coordinates were captured. The time between mouse coordinate captures depended on participant activity, but happened as soon as the host computer generated mouse events.

Every ten seconds, the owner process of all open windows was recorded. The process name captured was the name of the executable file, without the .exe extension. While capturing these processes, the window with focus was also recorded.

The software prompted users (Figure 3.6) to fill out a questionnaire (Figure 3.7) about their emotional status every hour. The questionnaire asked users to indicate how they felt using a five-point Likert scale. To ensure that the participants were actively using their computers, the questionnaire only appeared if the system recorded at least 2000 mouse and/or keyboard events over the previous five minutes. If less than 2000 mouse and/or keyboard events occurred during last five minutes, the software waited until the threshold was reached, then prompted the user to complete the questionnaire. As indicated from the previous section, users always had the option of skipping the questionnaire.

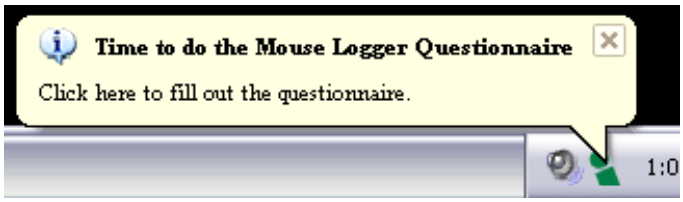


Figure 3.6 - Prompt for users to fill out questionnaire appeared approximately every hour depending on user activity

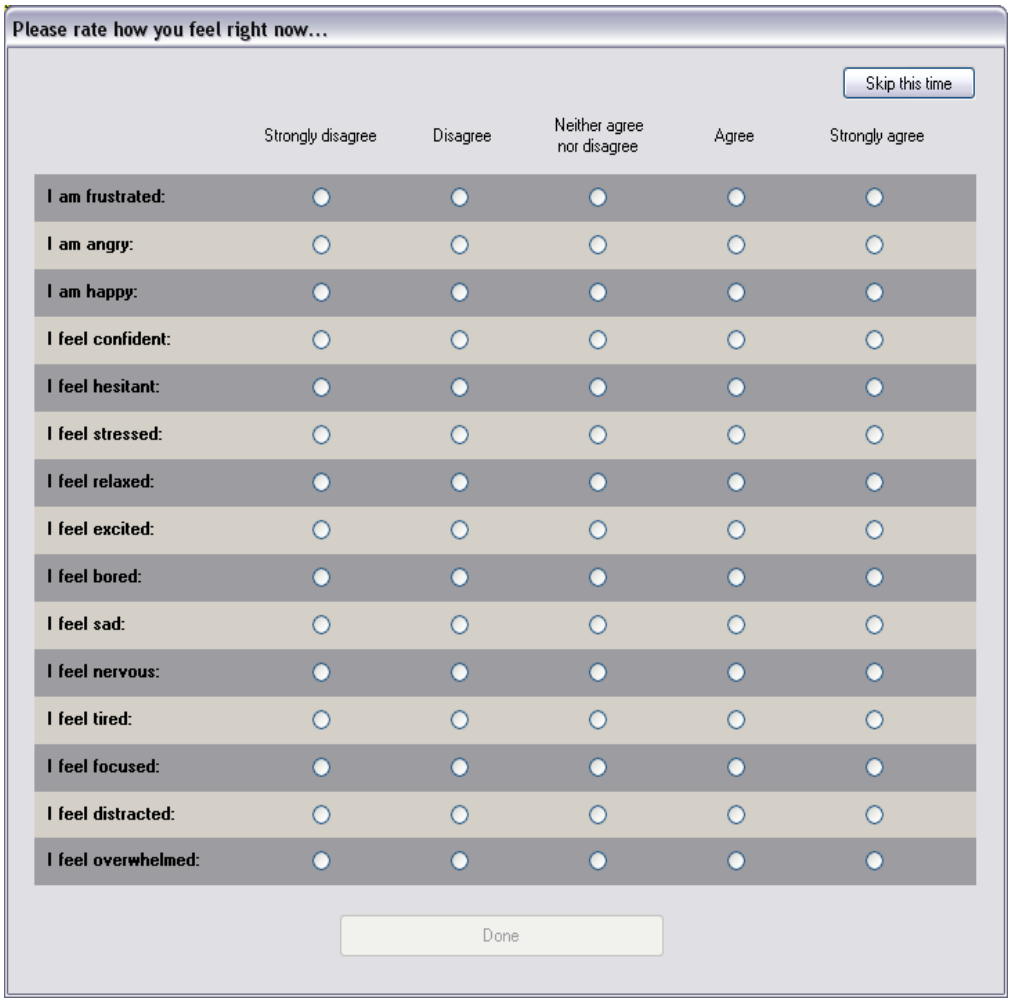
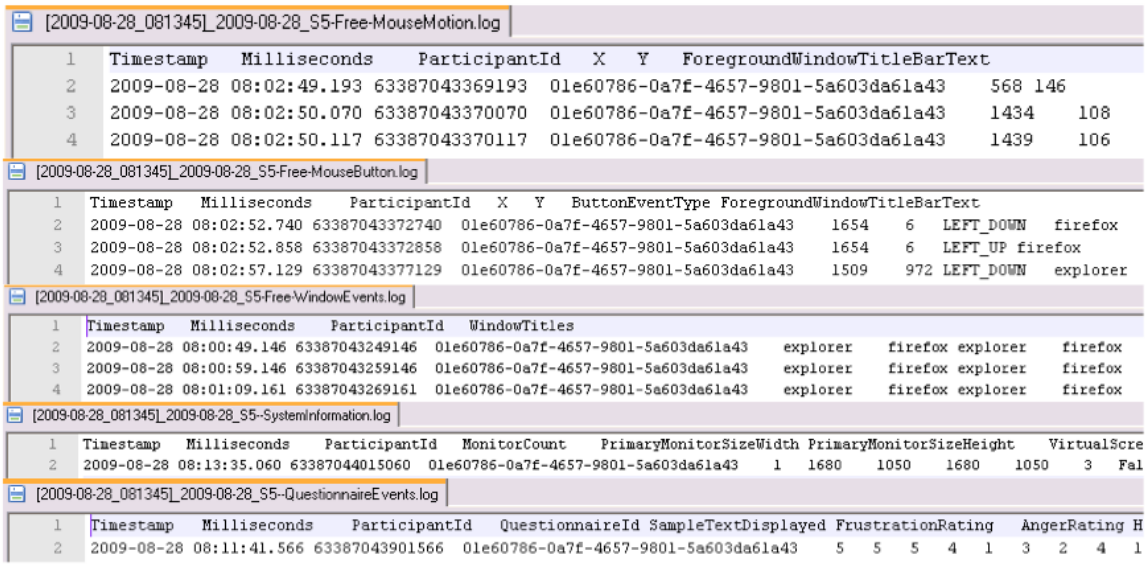


Figure 3.7 - Emotional states questionnaire

After each questionnaire, the software captured various system parameters that were later used to extract features. Parameters collected included monitor count, primary monitor resolution, virtual screen resolution, mouse button count, whether mouse buttons were swapped, mouse speed, mouse wheel present, double click time, double click dimensions, and drag size dimensions. Whether mouse buttons were swapped was important to know for determining the primary button used for single and double click operations. Double click and drag size dimensions were important for detecting double and single clicks respectively.

The software generated log files (see Figure 3.8) for seven different event types: keystroke, mouse button, mouse motion, mouse wheel, open windows, system configuration, and questionnaire. New files for each type were created each time the emotional states questionnaire was completed and were uploaded to a central server where they were available for analysis. Files waiting for upload were moved to the "Sending" folder. Each file was individually sent to the server and if successful, was moved to the "Sent" folder. If the sending operation failed, the files remained in the "Sending" folder until the next sending attempt. This ensured the client continued to attempt to send files to the server in case the server was unreachable. Thus, the client may have delivered files later than expected, but they would eventually reach the server when the server was again reachable.



[2009-08-28_081345]_2009-08-28_S5-Free-MouseMotion.log

```
1  Timestamp   Milliseconds    ParticipantId   X   Y   ForegroundWindowTitleBarText
2  2009-08-28 08:02:49.193 63387043369193  01e60786-0a7f-4657-9801-5a603da61a43    568 146
3  2009-08-28 08:02:50.070 63387043370070  01e60786-0a7f-4657-9801-5a603da61a43    1434    108
4  2009-08-28 08:02:50.117 63387043370117  01e60786-0a7f-4657-9801-5a603da61a43    1439    106
```

[2009-08-28_081345]_2009-08-28_S5-Free-MouseButton.log

```
1  Timestamp   Milliseconds    ParticipantId   X   Y   ButtonEventType ForegroundWindowTitleBarText
2  2009-08-28 08:02:52.740 63387043372740  01e60786-0a7f-4657-9801-5a603da61a43    1654    6   LEFT_DOWN   firefox
3  2009-08-28 08:02:52.858 63387043372858  01e60786-0a7f-4657-9801-5a603da61a43    1654    6   LEFT_UP firefox
4  2009-08-28 08:02:57.129 63387043377129  01e60786-0a7f-4657-9801-5a603da61a43    1509    972 LEFT_DOWN   explorer
```

[2009-08-28_081345]_2009-08-28_S5-Free-WindowEvents.log

```
1  Timestamp   Milliseconds    ParticipantId   WindowTitles
2  2009-08-28 08:00:49.146 63387043249146  01e60786-0a7f-4657-9801-5a603da61a43    explorer    firefox explorer    firefox
3  2009-08-28 08:00:59.146 63387043259146  01e60786-0a7f-4657-9801-5a603da61a43    explorer    firefox explorer    firefox
4  2009-08-28 08:01:09.161 63387043269161  01e60786-0a7f-4657-9801-5a603da61a43    explorer    firefox explorer    firefox
```

[2009-08-28_081345]_2009-08-28_S5--SystemInformation.log

```
1  Timestamp   Milliseconds    ParticipantId   MonitorCount    PrimaryMonitorSizeWidth PrimaryMonitorSizeHeight    VirtualScre
2  2009-08-28 08:13:35.060 63387044015060  01e60786-0a7f-4657-9801-5a603da61a43    1   1680    1050    1680    1050    3   Fal
```

[2009-08-28_081345]_2009-08-28_S5--QuestionnaireEvents.log

```
1  Timestamp   Milliseconds    ParticipantId   QuestionnaireId SampleTextDisplayed FrustrationRating   AngerRating H
2  2009-08-28 08:11:41.566 63387043901566  01e60786-0a7f-4657-9801-5a603da61a43    5   5   5   4   1   3   2   4   1
```

Figure 3.8 - Log file formats for mouse motion, mouse button, window title, system configuration and questionnaire events

The Client Recording Software used GUIDs (Global Unique Identifiers) as participant identifiers. We used the Microsoft .NET Framework's functionality for providing GUIDs which are 128-bit identifiers that are supposed to be unique across computers. This allowed the unique generation of participant identifiers on our participants' computers without having to query our central server and ensured the anonymity of our participants. We implemented this in the Client

Recording Software by generating a GUID when the software ran for the first time and used the GUID as the participant identifier for all subsequent activity. Examples of a GUID can be seen in [Figure 3.8] – 01e60786-0a7f-4657-9801-5a603da61a3.

3.4 Web Service

A Java-based web service was created to receive log files from the client software. The web service was deployed in a Tomcat Java web container installed on the Interaction Lab's server. A single web method accepted the experiment name, participant identifier, file name, and the log file contents as base-64 encoded text. Base-64 encoding allowed transmission of text and binary data, although we only used text-based log files in our study.

The web service stored the log file data on the server's file system. The directory structure is shown in Figure 3.9. The top-level directory, named *data*, contained a list of directories for various experiments. Our study was titled *UofS_MouseKeyFieldStudy_1.2.1*. The experiment directory contained a directory for each participant and each participant's directory contained log files from that participant.

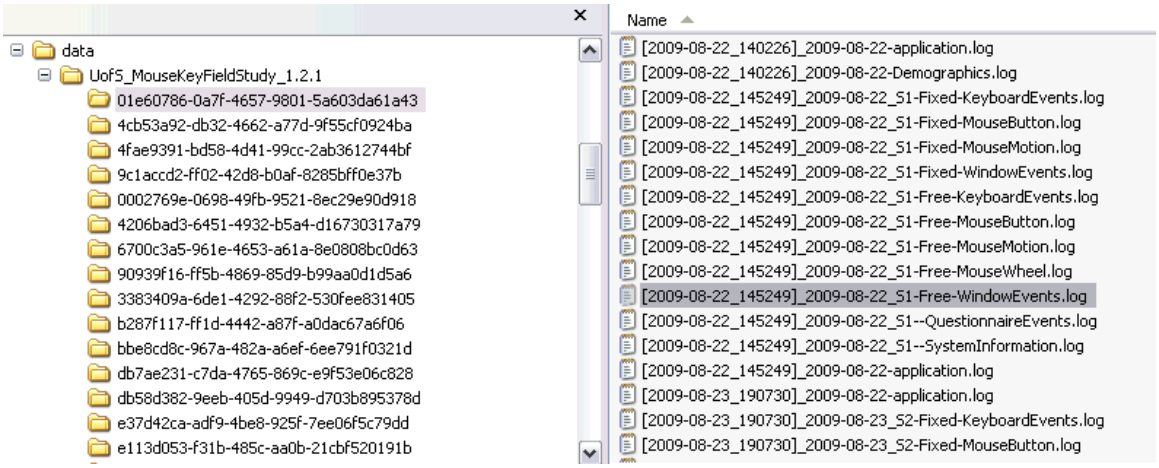


Figure 3.9 - Directory structure of log files stored on the server

When the web method was invoked, the web service used the experiment and participant names to locate the corresponding directory to save the log file. The log file contents were decoded from the base-64 encoding and saved to the participant's directory.

3.5 Data Retrieval Web Application

To provide easy access to our participant's log file and to monitor the progress while carrying out the study, we created a web application so researchers could view and download participants' log files. Our web application used the Ext GWT framework[2], a Java-based web application framework based on the GWT[3] (Google Web Toolkit) and the Ext JS framework[4]. With GWT, web applications are created much like a typical desktop application using Java Swing. The request/response nature of web applications is hidden by the toolkit and developers program event handlers for user interface events (e.g., button clicks) as though the web application were a desktop application. The Ext JS framework is a JavaScript framework that provides a large selection of user interface widgets for web applications. Ext GWT combines the GWT and Ext

[2] http://www.extjs.com/products/gwt/

[3] http://code.google.com/webtoolkit/

[4] http://www.extjs.com/

JS framework, providing a programming paradigm similar to desktop application development with a large selection of user interface widgets.

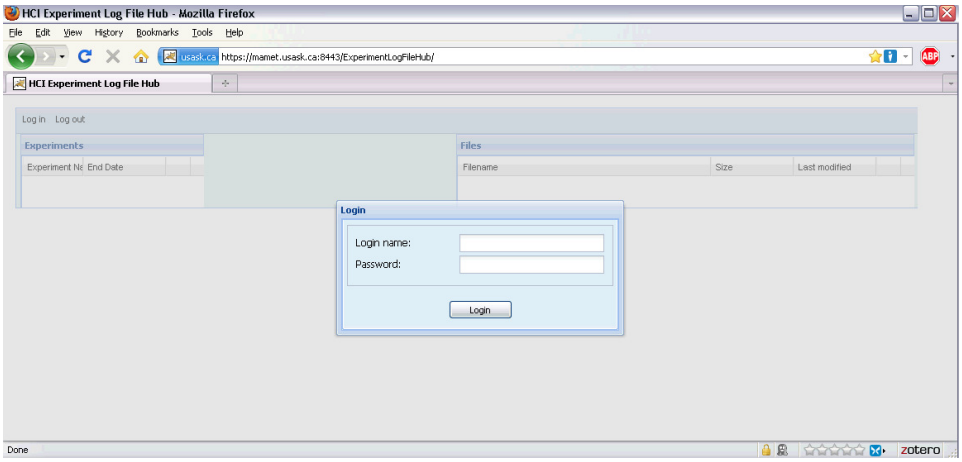


Figure 3.10 - The web application was password protected and used HTTPS to protect participant data

The data retrieval web application allowed us to do four things: view participant log files, download a participant's log files, download all of an experiment's log files, and set an experiment's end date. The web application was password protected (Figure 3.10) and allowed us to view an experiment, an experiment's participants, and all log files associated with the participant (Figure 3.11). It was important to view a participant in an experiment because it indicated the participant had installed the software and the software was running. It was important to view the log files for a participant because they provided an indication the software was operating properly. We found this useful when we had an issue in which some participants were erroneously asked to fill out the demographic questionnaire every time the study software started (it should have only asked the first time the software started). The software did not register that the demographic questionnaire was completed and the emotional state questionnaire would not appear. We were able to identify participants having this problem because the participant's folder would appear on the server indicating that the field study software was running, but the demographic log file did not appear. Furthermore, we noticed this only affected a small number of participants (two), so it was not a system bug that would require us to stop the

study and redesign the software. We chose not to fix this problem because it affected so few participants.

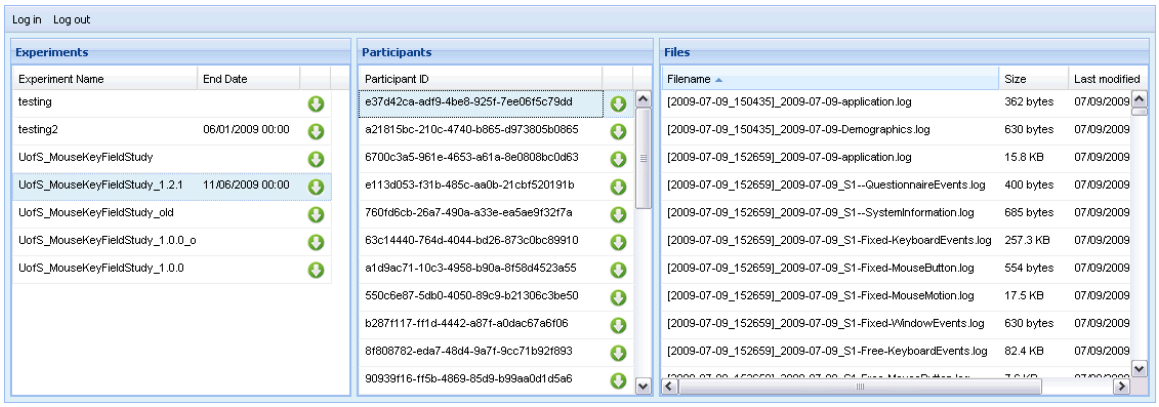


Figure 3.11 - Ext GWT web application for viewing, downloading and setting experiment end dates

When the download operation was invoked, the server created a zip archive of the files for the participant or the entire experiment (see Figure 3.12 and Figure 3.13). We could then download the log files as a single file rather than receive them separately. Downloading a single participant's log files or all participant log files was useful for evaluating our summarization scripts after the study began. There were cases when our summarization scripts were revised to accommodate cases we had not foreseen in the data. For example, some participants used applications with names that caused our MatLab scripts to fail. After identifying this problem, we wrote a different script to replace unsuitable characters in the application name string to ones that our MatLab scripts could use.

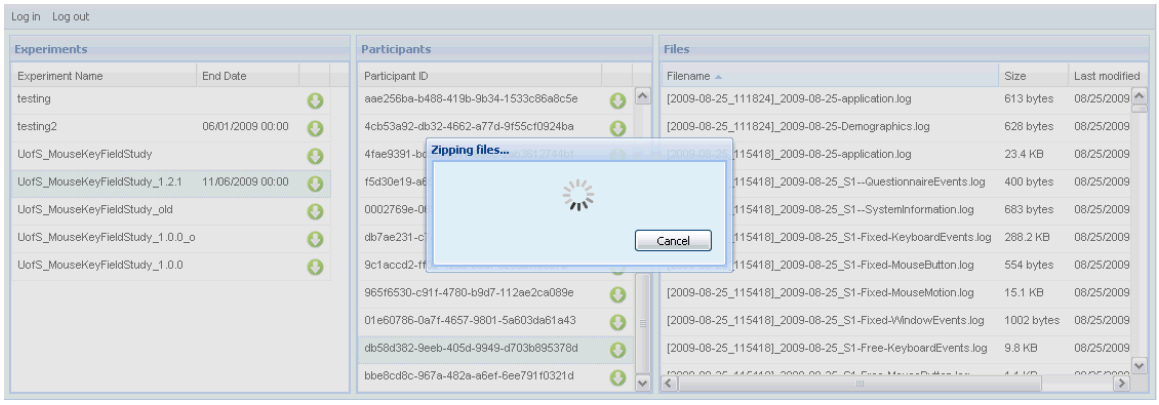


Figure 3.12 - Zip archives were created when we wanted to download experiment or participant log files. This could take several minutes.

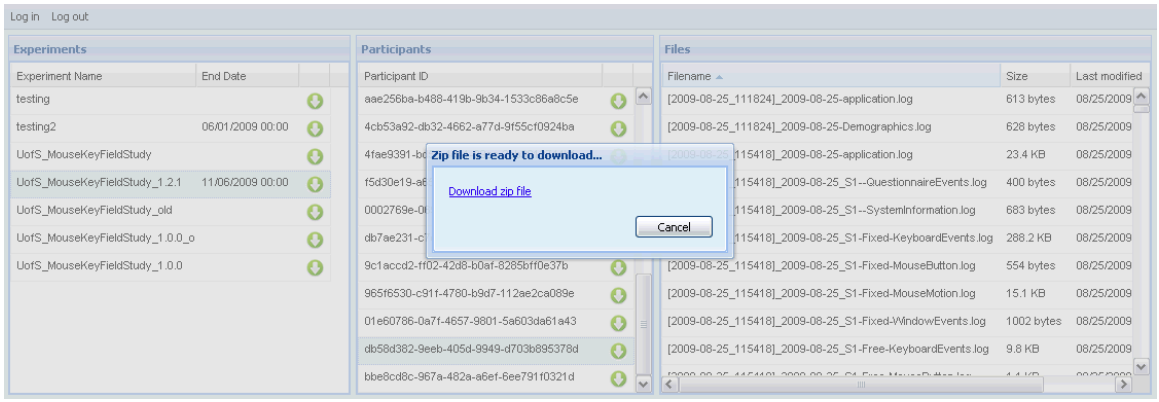


Figure 3.13 - After the zip archive was created, it was available for download.

Setting an end date for an experiment prevented the web service from accepting new log files after an experiment was finished (see Figure 3.14). This was important to ensure our study data was not corrupted after the field study was completed. This was useful when a pilot study we conducted finished and we wanted to prevent new data from arriving after we had asked our participants to uninstall the software.

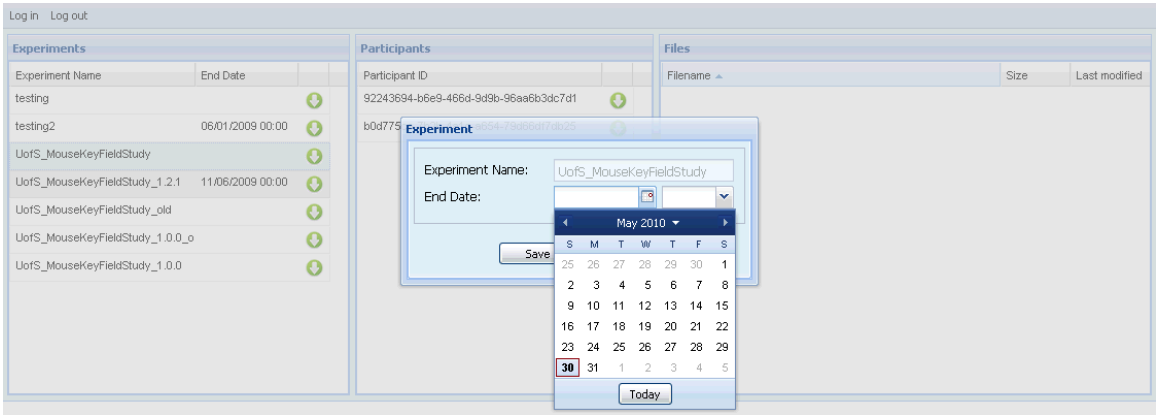


Figure 3.14 - Setting an experiment end date resulted in the web service rejecting new log files for the experiment.

3.6 Data Summarization

Data summarization was built to transform log file data into a format that could be analyzed using statistical and machine learning techniques. Because emotional states were captured using questionnaires, the variables associated with those emotional states also needed to be associated with the questionnaire responses. This was accomplished by summarizing mouse motion and button click data for the five-minute period before the questionnaire appeared. Summarizing only the data before the questionnaire appeared minimized potential bias caused by the appearance of the questionnaire and the participant answering the questionnaire.

A MATLAB script was used to summarize data for all participants. The script was provided the location of the unzipped experiment directory and produced CSV (comma separated value) files for each participant in which a row represented the results from a single questionnaire along with the summarized mouse motion and click variables. Each file contained all questionnaire results for a given participant and the files were concatenated later for analysis. The CSV format allowed easy conversion to the ARFF (Attribute-Relation File Format) used by the machine learning software Weka.

When processing a participant directory, the MATLAB script loaded all the participant's questionnaire, system, mouse button, and mouse motion events into MATLAB data structures. The calculations used for each mouse variable are shown in Table 3.5. The calculated variables for each participant were normalized so we could compare values across participants, i.e., inter-participant predictive modeling. We did a simple normalization calculated by dividing each calculated variable by the maximum value calculated for that variable and participant. For example, if participant P1 had a maximum left click count of 100, each normalized value was calculated by dividing the calculated value by the maximum value of 100:

$$leftClickCount_{norm}(i) = \frac{leftClickCount_i}{\max(leftClickCount)}$$

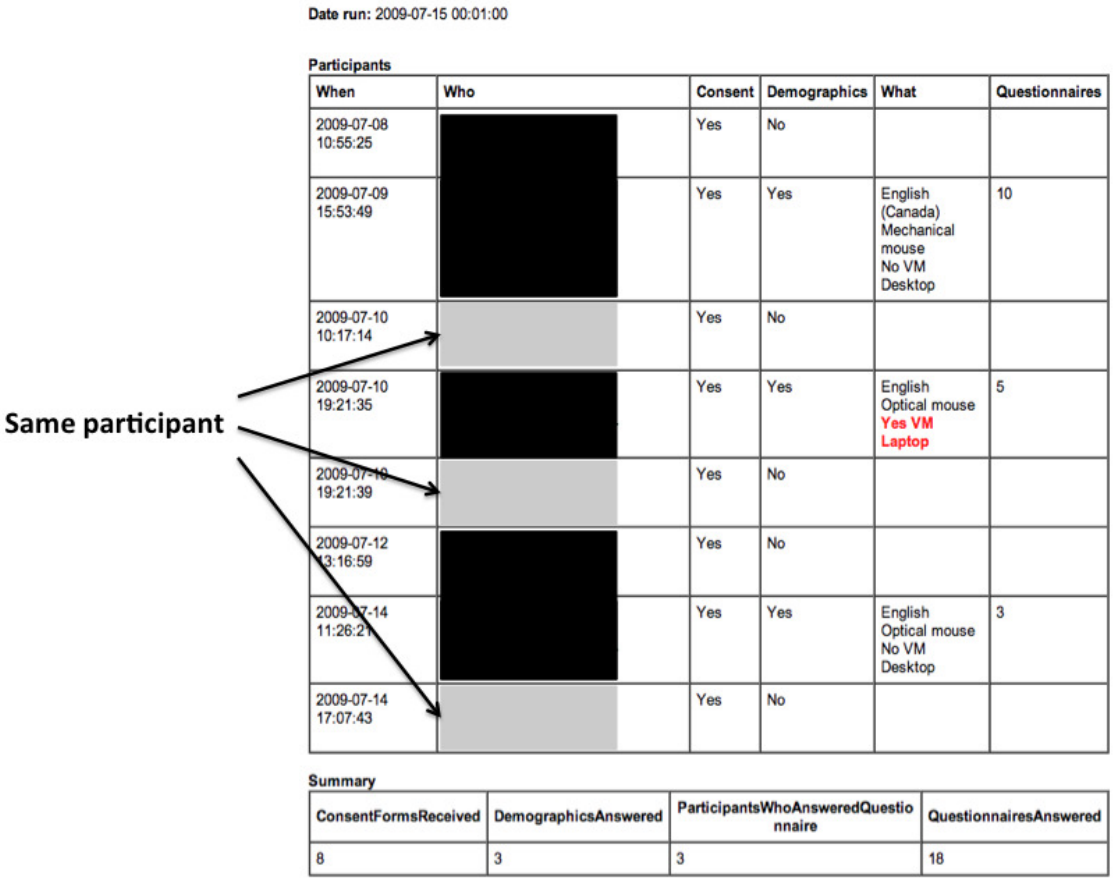
The normalization process was repeated for all the mouse variable calculations and for all participants.

Table 3.5 - Description of mouse variable calculations

Calculation	***Description***
Left click count	Count of left button up events
Right click count	Count of right button up events
Total click count	The sum of left clicks and right clicks
Single click count	Count of sequential left button down and up events where the mouse x and y coordinates have not changed more than the system setting for drag width and height and this is not part of a double click
Single click dwell time	Milliseconds between left button down and up events
Double click count	Count of sequential left button down, up, down and up events where the time between the first and second left button down events is less than the system setting for double click time and the change in mouse cursor x and y positions for the two button down events is less than the system setting for double click width and height respectively
Double click - first click dwell time	Milliseconds between the double click's first click down and up events
Double click - second click dwell time	Milliseconds between the double click's second click down and up events
Double click - time between first and second click	Milliseconds between the double click's first down and second down events
Total distance	The sum of the distances between each mouse cursor motion event where distance between two coordinates is calculated as the Euclidean distance: $$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$
Speed	Speed was calculated for two mouse cursor motion events as the change in mouse cursor location over the time difference between those two events: $$speed = \frac{d}{t_2 - t_1}$$
Acceleration	Acceleration was calculated for two mouse cursor motion events as the speed divided by the time: $$acceleration = \frac{speed}{t_2 - t_1}$$
Jerk	Jerk was calculated for two mouse cursor motion events as the acceleration divided by the time: $$jerk = \frac{acceleration}{t_2 - t_1}$$
Still moment count	The number of times the duration between mouse cursor motion events was greater than 1 second
Still moment duration	Sum of the duration of still moments

3.7 Daily Report

Daily report scripts were used to gauge participant activity during the field study. They provided a summary of completed consent forms, completed demographic forms, system configurations (e.g., desktop, laptop, optical mouse, language) and completed questionnaires (see Figure 3.15).



Date run: 2009-07-15 00:01:00

Participants

When	Who	Consent	Demographics	What	Questionnaires
2009-07-08 10:55:25	[redacted]	Yes	No		
2009-07-09 15:53:49	[redacted]	Yes	Yes	English (Canada) Mechanical mouse No VM Desktop	10
2009-07-10 10:17:14	[redacted]	Yes	No		
2009-07-10 19:21:35	[redacted]	Yes	Yes	English Optical mouse **Yes VM** **Laptop**	5
2009-07-10 19:21:39	[redacted]	Yes	No		
2009-07-12 13:16:59	[redacted]	Yes	No		
2009-07-14 11:26:21	[redacted]	Yes	Yes	English Optical mouse No VM Desktop	3
2009-07-14 17:07:43	[redacted]	Yes	No		

Summary

ConsentFormsReceived	DemographicsAnswered	ParticipantsWhoAnsweredQuestionnaire	QuestionnairesAnswered
8	3	3	18

Figure 3.15 - A daily email indicating participant progress in the study (names and email addresses have been removed).

The daily emails were also available from a web page (Figure 3.17) that was secured using the University of Saskatchewan's Central Authentication Service (CAS)[5] (Figure 3.16).

[5] http://www.usask.ca/docs/cas/

Figure 3.16 - Central Authentication Service (CAS) login screen

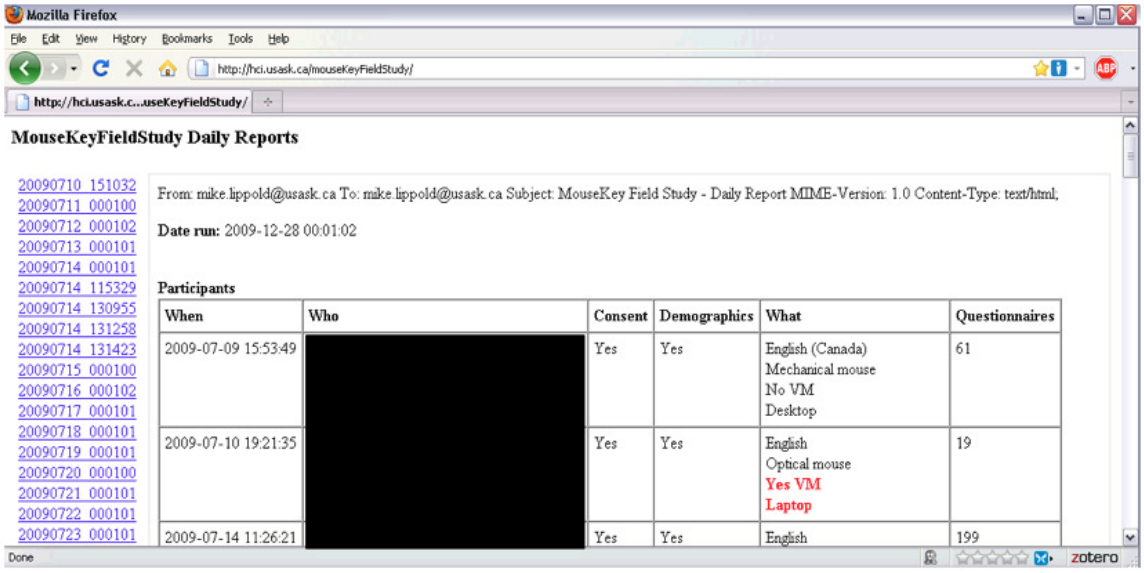


Figure 3.17 - Daily reports were available from a password protected web site (names and email addresses have been removed).

Both the emails and web site were useful for identifying early problems with the client software. As mentioned, we noticed some participants had difficulty in filling out the demographic questionnaire. This problem is evident in Figure 3.15 in which one participant can be seen to have installed the software three times (each line indicates a new installation of the software), but

did not successfully complete the demographic questionnaire. We were able to follow up with the participant and resolve the issue.

Further details of the software are described in Appendix C. In the next chapter, we will present how the ESM software was deployed in a field study.

4 FIELD STUDY

In the previous chapter we described software for capturing subjective emotional state labels and mouse data for a field study. To evaluate our approach and architecture, a field study was conducted from July to October 2009. In addition to the data described in Chapter 3, we also collected keystroke data to support another research project.

In this chapter, we describe our field study including: the participants; the hardware and software requirements of participants' computers; the procedure used to engage and disengage participants and monitor overall status of the data collection; and our analysis process. We begin by describing the participants we recruited for our study.

4.1 Participants

Twenty-seven participants, twenty-one male and six female, were recruited for the study (see Figure 4.1). We targeted between 20 and 30 participants which is consistent with other studies that have performed predictive modeling based on mouse dynamics [37,38]. Participants ranged in age from 18 to 59 with a mean of 27.6 years. All participants were right-handed mouse users. Thirteen participants installed the study software at home, thirteen at work, and one at another location. Nineteen participants used their computers more than 4 hours per day and twenty-three spent more than 50% of their time on the computer the software was installed on.

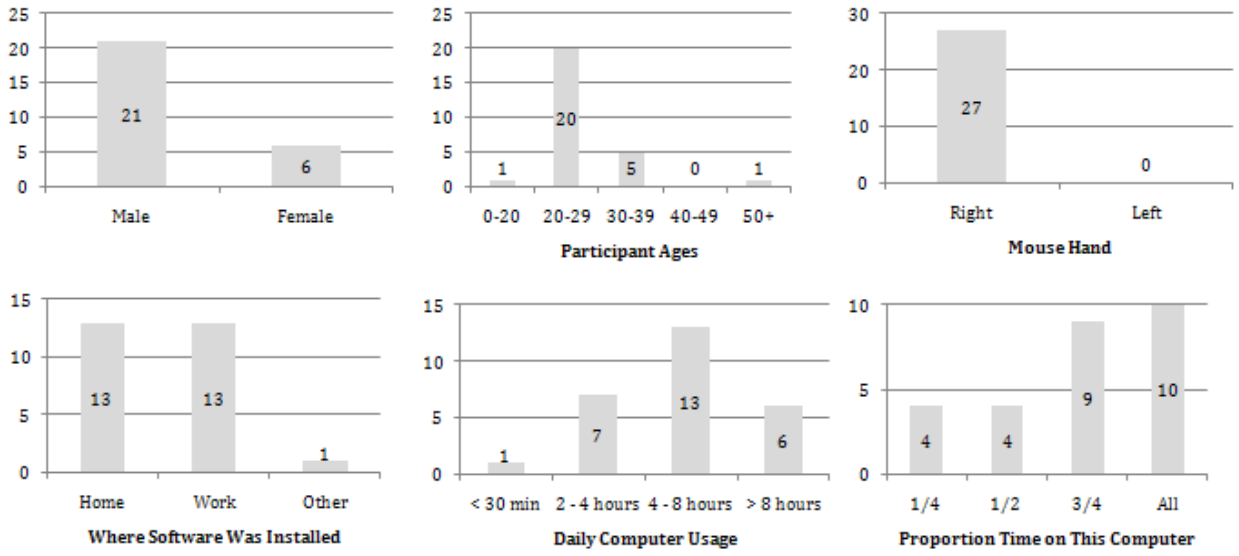


Figure 4.1 - Participant demographic information. The y-axis represents the number of participants in each category.

4.2 Apparatus

Participants were required to install the experimental software described in Chapter 3. The software was designed to work on Windows XP and Vista operating systems, so participants' computers required one of those two operating systems. Participants could have multiple monitors (X had one monitor and Y had two monitors).

Only data from *mouse-only computers* – computers whose only pointing device is a mouse – were used for the predictive modeling described in Chapter 5; participants whose computers had a track pad or combination of mouse and track pad were excluded. Fourteen participants had mouse-only computers. We did this because different pointing devices have different mechanical requirements on the human anatomy. Detecting small jittery motions on two different devices could be very different. Therefore, we decided to remove this potential confounding factor.

4.3 Procedure

Participants were solicited by two main methods. We advertised our study on bulletin boards across the University of Saskatchewan campus. The second method was advertising on the University of Saskatchewan's PAWS web site[6]. Potential participants were forwarded to our field study web site which explained the study and the need to install software that logged their mouse activity. Visitors were required to consent before they could download the logging software. A consent form web page was presented to users where they had to check a box indicating their consent to participate in the study. After visiting the download page once, visitors could regain access to the download page without downloading the consent form again.

The software performed as described in Chapter 3. After the software was installed, participants were prompted to fill out a demographic questionnaire, after which the software began its sampling function. At one-hour intervals, participants' mouse activity was recorded for five minutes after which the emotional state questionnaire appeared. Participants were not required to perform any mouse-pointing tasks as would occur in a laboratory study.

Participant data was collected for two months. This duration was an estimate of the length of time required to collect enough data from participants to model and was partially based on two pilot studies conducted. At the end of the two-month period, an email was sent to participants, thanking them for their participation and asking them to uninstall the software. At the end of the study in October, we provided all the participants with an honorarium to thank them for their participation.

4.3.1 Incentives for Participation

In addition to the honorarium, every week the three participants with the largest number of completed questionnaires were entered into a lottery that was held after the field study data collection period was complete. Three winners were randomly selected from the lottery pool and each received a one hundred dollar gift certificate. This was widely advertised to participants to

[6] http://paws.usask.ca

encourage them to answer the emotional state questionnaire as frequently as possible. Participants did not receive feedback on their progress relative to other participants or whether they were one of the top three in a particular week.

4.4 Method of Analysis

This section describes the process used for analyzing the mouse and questionnaire data. The MATLAB script described in Chapter 3 was invoked to generate a CSV file for each participant. Each file consisted of a row containing the results from a single questionnaire and the summary variables described in Chapter 3. All the participant files were concatenated for analysis.

Once the participants' summarized files were generated and aggregated into a single file, descriptive statistics were generated using Excel and SPSS. These gave us an idea of the distribution of participant answers and helped point us in directions for further analysis. For example, the distribution of ratings for most emotional states was broad for some ratings and narrow for others. This would cause severe class skew when creating our predictive models, so we decided to create a three-rating representation of emotional states where we collapsed 1 and 2 together, left 3, and collapsed 4 and 5 together. The resulting ratings were 1 for disagree, 2 for neutral, and 3 for agree.

4.4.1 Predictive Models

Predictive models were created and evaluated in a machine-learning tool called Weka [32] (for an explanation of predictive modeling, see section 2.6). Weka was used to filter data as well as create and evaluate models. For each emotional state, a predictive model was created where the three ratings were used as classes. Dimensionality reduction, reducing the number of features [2], is important because it simplifies the problem space being solved and a simpler problem space requires fewer instances for each class [75]. Dimensionality reduction was done using the PrincipalComponents (PCA) filter in Weka. PCA (principal component analysis) is a mathematical technique that attempts to reduce the number of variables by combining related variables while still accounting for some defined amount of variance in the data [3]. Weka's J48,

a type of decision tree algorithm based on a well-known decision tree algorithm called C4.5, was used to create the models. We used this over other algorithms because C4.5 allows missing values, performs classification into nominal categories, and models we created in a pilot studied showed better results using the J48 than others such as neural networks. 10-fold cross validation was used to build and evaluate each model.

The predictive models were created on three different samples of data sets: original, under-sampled, and 160 instances per class. The original data set consisted of the original instances collected. The under-sampled data sets were created by taking a sub-sample of those classes that had more instances than the minority class. For example, consider an original "Angry" data set with 10 instances of "disagree", 10 of "neutral", and 5 of "agree". To under-sample we would take a subset of disagree instances of size 5 and a subset of neutral instances of size 5. The final data set used would be 5 disagree, 5 neutral, and 5 agree instances for a total of 15 instances. We used Weka's "random subset selection" to create our under-sampled data sets.

The "160 instances per class" data sets were used based on the recommendation by Jain et al. [40], who suggest that 5- to 10-times the number of features for each class should provide enough data points to build a model. In our study, 10-times the number of features would be 160 instances. However, we did not have 160 features for every class in the inter- or intra-participant data sets, but we could achieve this using a combination of over- and under-sampling. We used Weka's Synthetic Minority Oversampling Technique (SMOTE) [15] implementation for over-sampling and random subset selection for under-sampling.

5 RESULTS

This chapter reports the results of the field study described in the previous chapter. We begin by presenting the response rate of the ESQ (emotional state questionnaire) from an overall perspective as well as at the participant level. Next, the distribution of ratings is reported, again from both the participant and overall perspectives. The data collection performance is reported with a brief discussion of possible techniques for improving it. And finally, we present the results of our predictive models.

5.1 ESQ Response Rate

In this section, we present results of participant response rates for the ESQ. We consider the following metrics related to response rate: days participants were active in the study, questionnaires requested, questionnaires completed, and questionnaire completion rate. The following subsections present more detailed results for the response rate metrics (see Figure 5.1 for per participant results).

5.1.1 Days Participants Were Active in the Study

We calculated the number of active days by counting the number of days spanning the first to the last emotional state questionnaire received. Participants were active in the study for a mean duration of 47.5 days, less than the two months period we asked people to participate in the field study. Sixteen participants actively answered questionnaires for 50 days or more and eight answered questionnaires for more than 60 days. If we use 50 days as the mark for completing the two month study period, 62% of participants finished the study. Three participants, 11.5% of all participants, were active for more than 70 days, one as high as 89 days (P17).

Two participants, P15 and P25, were active for only a single day indicating they stopped running the software. They may have uninstalled the software, gone on vacation, or otherwise stopped

using their computer. Unfortunately, we did not follow-up with them to discover why they stopped using the software. Those two participants, P15 and P25, were excluded from the predictive modeling process.

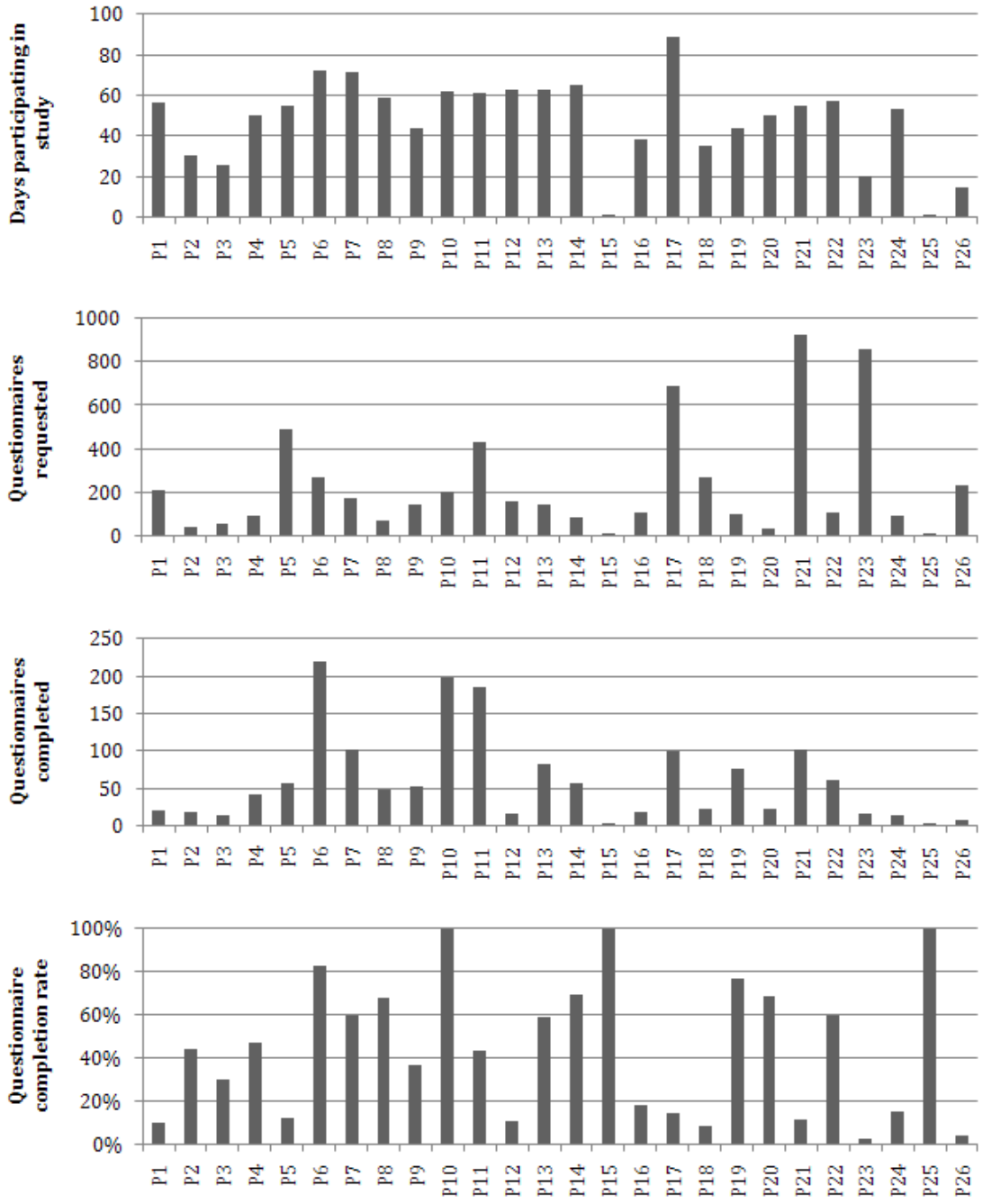


Figure 5.1 – Participant questionnaire activity during field study

5.1.2 **Questionnaires Requested**

Questionnaire requests were calculated based on the number of times the task bar balloon appeared asking users to complete the ESQ. If the taskbar balloon was ignored, either because the participant did not see it or intentionally ignored it, it would disappear after 60 seconds and re-appear. Each re-appearance was also counted as a request. The total number of ESQ requests was 5950, but there were considerable differences in the number of requests for each participant. We explain individual differences using three factors: duration in study, participants ignoring requests, and mouse/keyboard activity.

Length of duration in the study was one reason for a participant having a large number of requests. Participants who were only in the study for a short time, such as P15 and P25 (one day each), had a low number of requests compared to those who were active in the study for longer periods, such as P17.

Ignoring ESQ requests, either because users did not see the task bar balloon or intentionally ignored it, was also a factor influencing the number of requests. P23 frequently ignored requests (Figure 5.2), but another participant, P26 also had a large proportion of ignored requests yet comparatively few overall requests. P23 and P26 were in the study for 20 and 15 days respectively, yet P23 had a much higher request rate (858) compared to P26 (232).

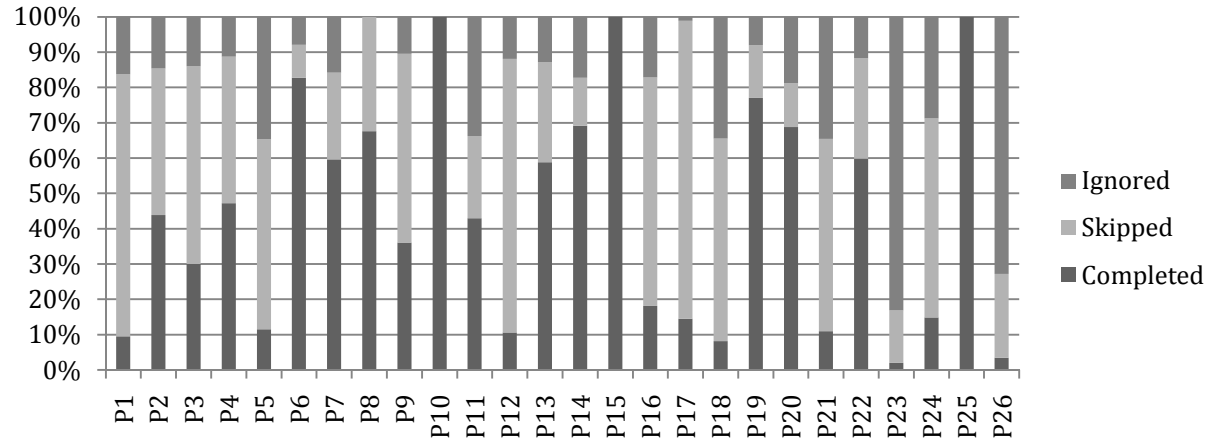


Figure 5.2 - Proportion of questionnaires ignored, skipped and completed per participant

Recall from Chapter 3 that the software required a threshold of mouse and/or keyboard activity for the software to prompt users to complete the ESQ. We attribute the differences between P23 and P26 to the activity threshold. P21 and P24 have similar proportions of ignored requests; yet again the request counts are quite different. Again, we attribute these request differences to activity differences between the users. In fact, P21 was considerably more active than all other users with an average of 16.7 requests per day and log files indicate this participant used their computer much more than the other participants.

5.1.3 Questionnaires Completed

Completed questionnaires are the number of questionnaire responses we received from participants. 1,555 questionnaires were completed over the duration of the field study. Figure 5.1 shows the number of completed questionnaires by participant.

5.1.4 Questionnaire Completion Rate

Completion rate was calculated by dividing the number of completed questionnaires by the number requested. The mean completion rate was 26.2 percent. The extremely high completion rates of P15 and P25 (both 100%, see Figure 5.2) occurred because both only participated for one day and completed all questionnaire requests. P10 completed all 199 questionnaire requests, but unfortunately much of their data is useless for modeling because they did not vary their response (Figure 5.3). Similarly, P11 did not vary their responses. P1 (9.6%), P5 (11.5%), P12 (10.6%), P18 (8.1%), P21 (11%), P23 (2%) and P26 (3.4%) had response rates well below the mean. P1 participated in the study for 56 days, answering 20 of 209 questionnaires, and only had 34 ignored requests. P5 participated in the study for 55 days, answering 56 of 485 questionnaires, and had 168 ignored requests. P12 was in the study for 63 days, responded to 17 of 160 questionnaires, and only had 19 ignored requests. P18 participated for 35 days, only answering 22 of 270 questionnaires with 93 ignored requests. P21 answered 101 of 921 over 55 days and had 318 ignored requests. P23 only answered 17 of 858 with 713 timeouts over 20 days. P26 responded 8 times out of 232 over 15 days with 169 ignored requests. We discuss issues related to ESQ response rate in Chapter 6.

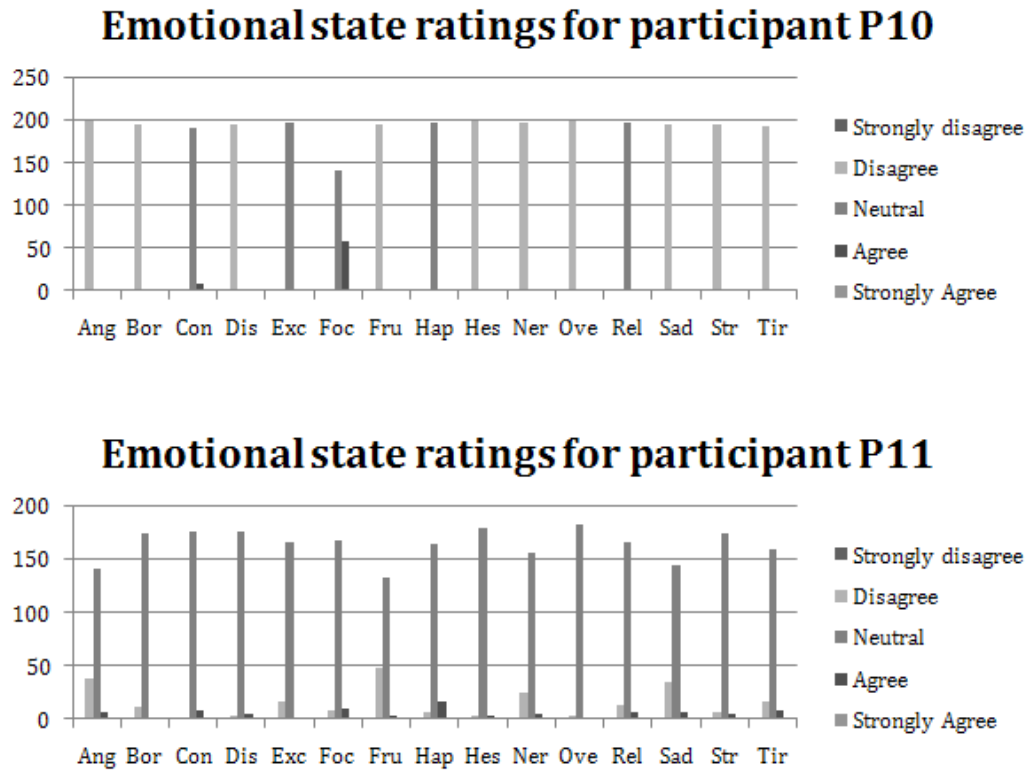


Figure 5.3 - Emotional state ratings for participants P10 and P11

In the next section, we report our findings related to the distribution of ratings participants gave when answering question in the ESQ.

5.2 ESQ Ratings

In the previous section, we reported the results of two indicators of success of our approach: complete rate and number of questionnaires completed. Another success indicator is whether the ESQ data can be used for predictive modeling and this is dependent on having a certain minimum number of ratings for each class of the predictive model. In this section, we report the results of participants' ESQ ratings in the context of requirements for inter- and intra-participant predictive modeling.

Participants did not tend to answer at the extremes of the five-point scales, strongly disagree and strongly agree (Figure 5.4). For predictive modeling, this is a problem because machine learning algorithms require sufficient data in all classes to learn to distinguish between them. Using the five-point scale, none of the emotional states have a minimum rating count greater than 160 which is the number of class instances for an instance-to-feature ratio of ten [40]. This class skew was also observed in a pilot study before the field study and a solution was devised by rescaling the five-point scale into a three-point scale: strongly disagree and disagree became disagree, neutral remained neutral, and agree and strongly agree became agree. Rescaling from 5-points to 3-points is valid because the scales are ordinal. The results of rescaling can be seen in Figure 5.5.

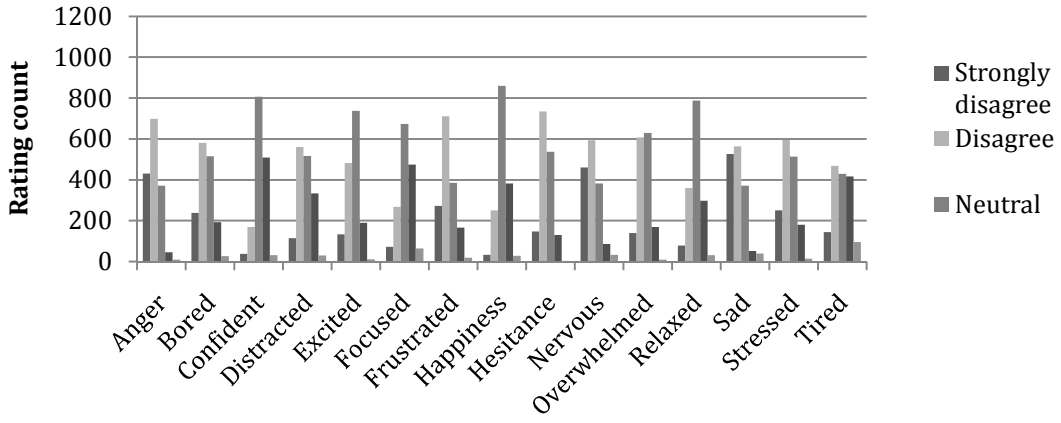


Figure 5.4 - All participants' emotional state ratings using 5-point scale

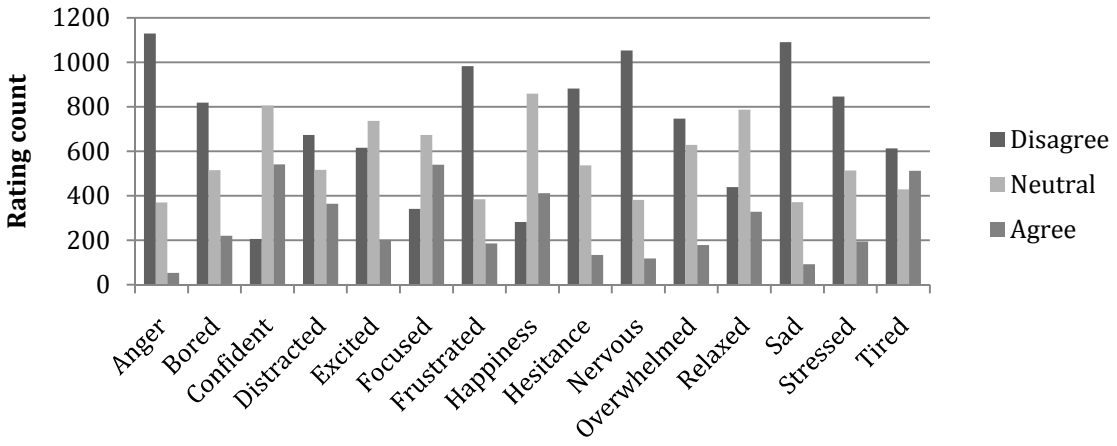
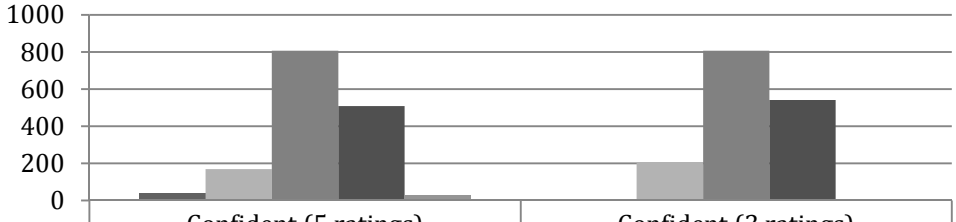


Figure 5.5 - All participants' emotional state ratings using rescaled 3-point scale

After rescaling, the ratings for most emotional states became more evenly distributed and eleven of the fifteen emotional states had enough class instances to meet the instance-to-feature ratio of ten. For example, the Confident emotional state had strongly disagree and strongly agree rating counts well below 160 -- 38 and 32 respectively (see Figure 5.6). However, after combining the disagree ratings with each other and the agree ratings with each other the resulting rating counts are well above 160 -- 207 and 541. The disagree and agree counts did not increase much, but two classes for which little data exists were eliminated – two classes that a predictive model no longer needs to consider and be trained for.



	Confident (5 ratings)	Confident (3 ratings)
Strongly disagree	38	
Disagree	169	207
Neutral	807	807
Agree	509	541
Strongly agree	32	

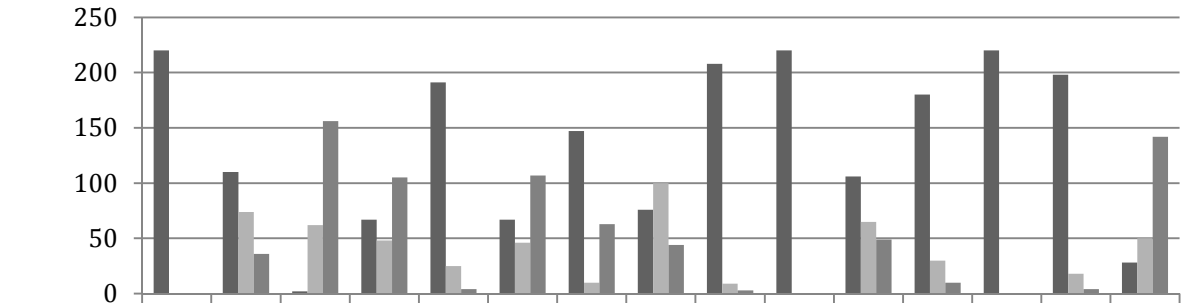
Figure 5.6 - Comparison of Confident 5- and 3-scale rating results

Collapsing from the five- to three-point scales did not improve the rating distribution for all emotional states and four did not reach the 160 instance threshold. The Anger emotional state is one example. Combining strongly disagree (431) and disagree ratings (699) resulted in 1130 instances, but the combined agree (45) and strongly agree (9) rating was 54, far below 160 (Figure 5.7).

	Anger (5 ratings)	Anger (3 ratings)
Strongly disagree	431	
Disagree	699	1130
Neutral	371	371
Agree	45	54
Strongly agree	9	

Figure 5.7 - Comparison of Anger 5- and 3-scale rating results

While rescaling increased the class instance counts to a suitable level for building models across participants (inter-participant) for eleven out of fifteen emotional states, it did not improve instance counts sufficiently for individual participant (intra-participant) models. P6 provided the most promising ESQ ratings for intra-participant modeling, particularly for the Bored, Distracted, Focused, Happy, and Overwhelmed states because they had the highest minimum rating counts (see Figure 5.8) compared to other participants. Yet the minimum rating counts were still below 160 instances.



	Ang	Bor	Con	Dis	Exc	Foc	Fru	Hap	Hes	Ner	Ove	Rel	Sad	Str	Tir
Disagree	220	110	2	67	191	67	147	76	208	220	106	180	220	198	28
Neutral	0	74	62	48	25	46	10	100	9	0	65	30	0	18	50
Agree	0	36	156	105	4	107	63	44	3	0	49	10	0	4	142

Figure 5.8 - Participant P6's 3-scale ratings

Overall, the quantity and quality (distribution of ratings) of ESQ responses were suitable for inter-participant modeling, but insufficient for intra-participant modeling. We discuss issues related to ESQ ratings in Chapter 6. Next, we present the results from the overall data collection capability of the software system.

5.3 Data Collection

In this section, we report results of the data collection capacity of our software and discuss some of the issues related to technological decisions that may have limited its capacity.

Over the course of the field study, 2.1 gigabytes of log files were collected. The Client Recording Software sent a package of files to the server after every questionnaire and whenever the application started. These files contained the ESQ response, mouse and keyboard data, application windows opened, and the client recording software's log files for the ten-minute period before the questionnaire appeared. The average size of file packages received was 1.42 megabytes and the majority of the space was keyboard and mouse data (>90%).

There are limits on the amount of data our implemented system can process. In a pilot study conducted before the field study, we recorded all mouse activity for the day and sent it from the Client Recording Software to the Data Collection Web Service at the end of the day. This

worked for most participants; however we encountered problems when the size of the files exceeded 20 megabytes. The Data Collection Web Service was a Web Service and used XML to wrap messages between it and the Client Recording Software. XML was handled using an XML processor which used the DOM (Domain Object Model). One drawback of the DOM is it loads the entire document into memory before it can be used and this uses much memory. In our case, the Data Collection Web Service ran out of memory. This upper limit can be increased by increasing the amount of memory in our server computer. A better solution is to switch to a less memory intensive representation of our files such as JSON (JavaScript Object Notation).

Establishing an upper limit for our system is important because it limits other types of probes that can be used. For example, facial gesture recognition is considered one of the more accurate techniques for determining user emotional states with recognition rates as high as 98% [26] and this could be accomplished by collecting images of participants' faces. The Data Collection Web Service's current limit of 20 megabyte file packages would not limit the overall system's capability of receiving images of participants' faces. Combining various probes may pose a problem, but as long as the overall size of data is less than 20 megabytes, the existing system would work.

We discuss the issues related to data collection in the next chapter. Next, we report the results of our predictive models.

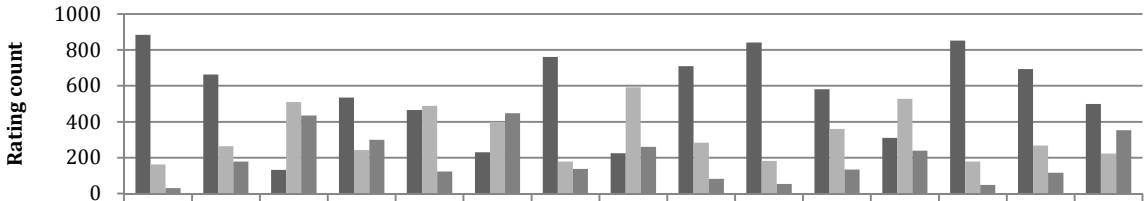
5.4 Predictive Models

The ultimate success of our approach is dependent on the creation of models that can predict the emotional state of participants. For the approach to be successful, the field study evaluation must show that predictive modeling features can be determined from the data. In this section, we show that the features extracted from the data were sufficient for predictive modeling, but that the combination of features selected did not provide predictive ability of the fifteen emotional states.

As mentioned in Chapter 4, only fourteen participants' data was used for predictive modeling. Our justification for eliminating certain participants (as discussed in Chapter 4) was to limit

pointing devices to computer mice because our models were created from inter-participant data (data from different participants) and we wanted to avoid possible device-dependent factors. While we would have liked to build intra-participant models, there was insufficient data to do so. For intra-participant models, we would have removed the device restriction.

For the fourteen participants, the rating counts are shown in Figure 5.9. A total of 1077 instances were used. Bored, Distracted, Focused, Happiness, Relaxed, and Tired were the emotional states that had rating counts greater than 160 for all three ratings, but we built models for all fifteen emotional states.



	Anger	Bored	Confidence	Distracted	Excited	Focused	Frustration	Happiness	Hesitance	Nervous	Overwhelmed	Relaxed	Sad	Stress	Tired
Disagree	884	663	132	535	465	231	760	225	710	841	581	310	851	694	499
Neutral	162	265	510	243	489	399	179	591	284	182	361	528	178	267	224
Agree	31	179	435	299	123	447	138	261	83	54	135	239	48	116	354

Figure 5.9 - 3-class ratings for the fourteen participants used for predictive modeling

The models with prediction rates greater than sixty percent are shown in Table 5.1 (the results of all models are shown in Appendix B). These seem like they performed well, but the Kappa values are relatively low for them all. While the prediction rates are much better than random chance (which would be thirty-three percent for a 3-class prediction model) the Kappa values are relatively low, which indicates the models did not account well for a chance algorithm that knows the distribution of classes.

Table 5.1 – Results for predictive models with a prediction rate greater than 60%

Data set	Test		Confusion matrix
	Prediction Rate	Kappa	
Anger	83	0.29	a b c <-- classified as 846 38 0 | a = 1 112 49 1 | b = 2 27 4 0 | c = 3
Sad	79	0.21	a b c <-- classified as 806 41 4 | a = 1 131 46 1 | b = 2 48 0 0 | c = 3
Nervous	75	0.10	a b c <-- classified as 788 46 7 | a = 1 151 21 10 | b = 2 47 3 4 | c = 3
Frustration	71	0.22	a b c <-- classified as 698 40 22 | a = 1 115 57 7 | b = 2 120 7 11 | c = 3
Hesitance	68	0.25	a b c <-- classified as 624 72 14 | a = 1 172 108 4 | b = 2 70 10 3 | c = 3
Stress	64	0.19	a b c <-- classified as 600 71 23 | a = 1 168 86 13 | b = 2 91 21 4 | c = 3

There are two interesting things to note here. First, the emotional states shown here are not among the six emotional states that had rating counts greater than 160 for all three ratings. This raises the question of whether there were enough instances of the minority class (the class with the fewest instances) to sufficiently train a model. Second, all six emotional states have one dominant class and the confusion matrices show that the models for these states highly favour that dominant class. Table 5.2 shows the most frequently occurring class is predicted more frequently than is actually occurring. For example, the Anger model classified 91% of all instances as "disagree", but only 82% of the instances were actually "disagree". By strongly favouring the majority class, the prediction rates are inflated and not indicative of how the model

would perform if the number of class instances were balanced. These models suffer from class skew.

Table 5.2 - Prediction rates for the dominant classes in models with overall prediction rates greater than 60%

	Model Predicted	Actual
Anger - disagree	91%	82%
Sad - disagree	91%	79%
Nervous - disagree	92%	78%
Frustration - disagree	87%	71%
Hesitance - disagree	80%	66%
Stress - disagree	80%	64%

The problem of class skew can be addressed by under-sampling the data set so that the number of instances of the more dominant classes matches the number of the least dominant class. The results for the six emotional states with all class instance counts greater than 160 are shown in Table 5.3. The first row shows the prediction rate and Kappa statistic for the original, whole data set. The second row shows the results for the under-sampled data sets which have no class skew because each class has the exact same number of instances. These models predict emotional states only slightly better than chance which would be prediction rate of 33 and have low Kappa values. This indicates that emotional states cannot be predicted from the mouse motion features used in our study.

Table 5.3 - Results of predictive models created on original and under-sampled data sets. Under-sampled data sets had a minimum of 160 instances in the least dominant class. Kappa statistic is shown in parentheses below the prediction rate.

Data set	Bored	Distracted	Focused	Happiness	Relaxed	Tired
Original	57% (0.13)	48% (0.13)	42% (0.09)	50% (0.07)	49% (0.14)	50% (0.20)
Under-sampled	38% (0.08)	42% (0.14)	38% (0.07)	45% (0.18)	44% (0.16)	40% (0.10)

The major problem with our modeling appears to be that the features do not contain information that the C4.5 algorithm can exploit to classify emotional states correctly. Despite this failure of the predictive modeling, the overall approach is successful because we were able to build models from our data in a related study [25] using the same software and field study data, did find some successful predictive models for emotional states. Furthermore, because the data collected is raw, un-interpreted data, in the future we can create new features based on the data and perform predictive modeling on them. This flexibility is an advantage of our approach because it allows researchers to analyze the raw field data after a field study. However, this flexibility can also be a disadvantage as researchers can continue to search for results indefinitely.

In the next chapter, we discuss issues related to the overall data collection approach and the poor performance of the models.

6 DISCUSSION

In this chapter, we discuss the results from the field study, identify areas of improvement for the future, and identify future work that follows from our research. In the Summary of Findings section, we discuss the results in the context of the quantity and quality of data provided by participants, ways of improving our predictive modeling, and the overall success of the data collection. In the section on Lessons Learned, we discuss ways of improving our study and future ESM studies by stressing the importance of identifying goals for the number of ESQs collected, motivating participants, and altering questionnaire frequency. Last, in the Future Work section, we discuss important follow-up work we believe should be undertaken.

6.1 Summary of Findings

Overall, we found our approach was successful. The field study demonstrated our approach's ability to collect ground truth data using the ESQ and collect data for and building predictive models. In this section, we summarize our findings of the data collection and predictive modeling.

We consider the results of our approach as it relates to the quantity and quality of data. We define quantity as the number of completed questionnaires. We define quality as the true representation of ground truth in completed ESQs and balanced ratings of emotional states. Figure 6.1 outlines the factors affecting quantity and quality. Next we explain the quantity factors.

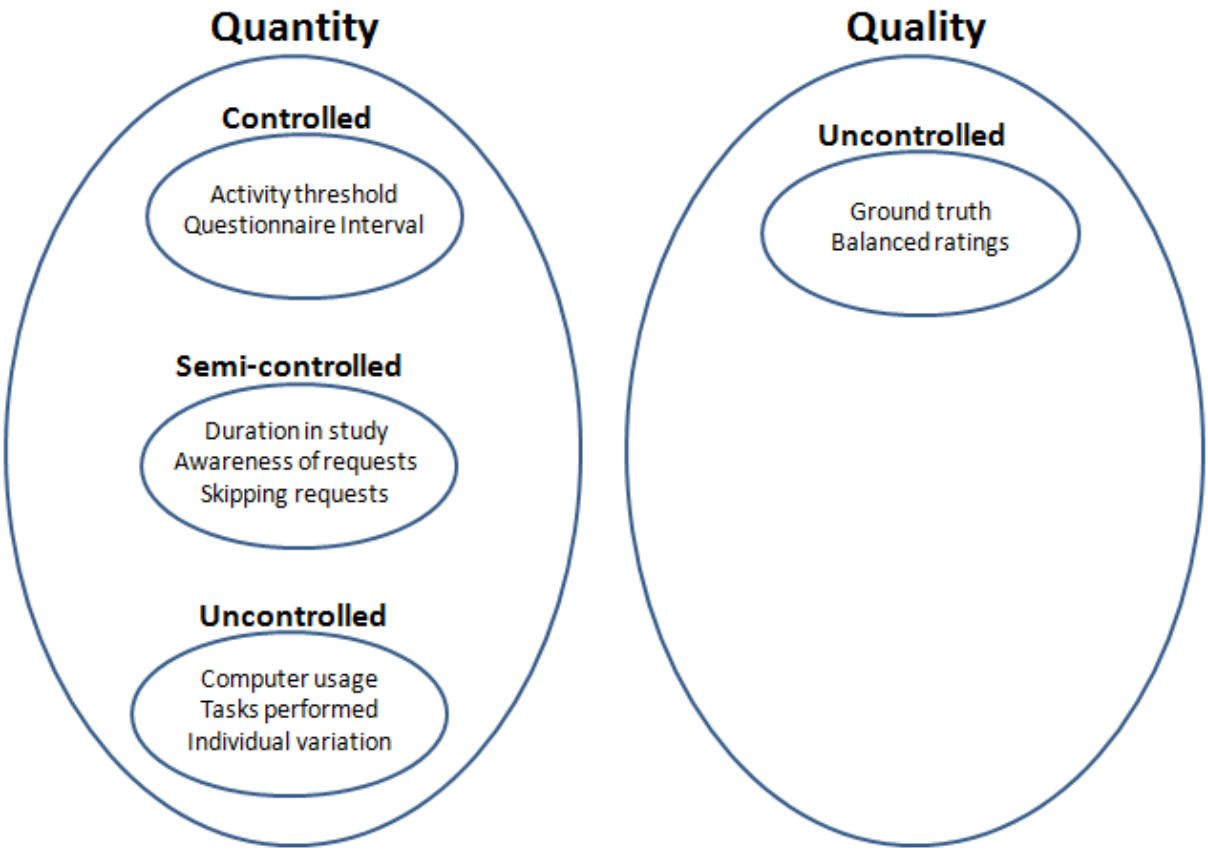


Figure 6.1 - Factors affecting quantity and quality of ESM data

6.1.1 ESQ Data Quantity

Our results showed that the quantity of each participant's completed ESQs varied from 3 to 220. There are a number of factors that affected the number of completed ESQs per participant. Controlled factors included activity threshold and questionnaire interval, both configured parameters in the Client Recording Software. Some factors are in a grey-area that may be indirectly influenced and we call these semi-controlled. Examples of semi-controlled factors are duration in study, and awareness of and skipping of requests. And finally, other factors are uncontrolled, i.e., the number of hours per day a participant uses their computer, tasks performed on the computer, and variation in how the same task is performed (e.g., clicking a menu item versus using a keyboard shortcut). We discuss these three types of quantity factors and how they affected the number of completed ESQs in this section.

6.1.1.1 Activity Threshold

Reducing the activity threshold could result in more ESQ requests, hence more ESQ completions. In Chapter 3, we described the algorithm for determining whether enough activity had occurred to create a sample and request participants fill out the ESQ. Based on testing before the field study, we chose a combined count of 2,000 mouse and keyboard events over a five minute period as the minimum threshold for a sample. If a participant's activity did not meet that threshold, the questionnaire would not appear. The 2,000 event threshold appeared to be a reasonable balance between ensuring enough mouse and keyboard data was captured and participants being asked to complete the ESQ. However, reducing the threshold could result in participants being asked more frequently to complete the questionnaire.

6.1.1.2 Questionnaire Interval

Decreasing the time between questionnaire requests could also increase the number of ESQ completions. The Client Recording Software waited one hour after an ESQ was completed to request another. There is a balance between annoying participants by asking them too frequently to complete a questionnaire and getting as many completed ESQs as possible. Unfortunately, we did not ask participants whether they perceived the ESQ requests to be occurring too frequently.

6.1.1.3 Duration in Study

The duration participants were active in the study limits the number of questionnaires they would be requested to complete. We were clear at the beginning how long participants were to be active in the field study, but we still had a variation in the number of days participants were active. Some were far below our target of 45 days and this may have been a matter of the incentives not making the perceived amount of effort (or distraction) worthwhile for those participants. Still, 62% of participants were active for more than 45 days. Increasing the duration of the study to be longer (e.g., one year) would help us to gather significant amounts of data from active users. An alternative approach would be to alter the incentives for participation, which will be discussed in 6.2.2.

6.1.1.4 Computer Usage

Greater computer usage generally results in more questionnaire requests. Recall that the software was designed to request participants to complete the ESQ every hour, if threshold number of mouse/keyboard events was reached. For example, a participant working on their computer for eight hours could be requested to complete the ESQ up to eight times whereas a participant using their computer for four hours could be prompted up to four times to complete the ESQ. Thus, participants who used their computers for longer durations in the day were more likely to be requested to answer the questionnaire more frequently.

6.1.1.5 Tasks Performed

The tasks participants' performed may have affected the number of ESQ requests. In Chapter 3, we described that users were only requested to complete a questionnaire if they had reached a minimum threshold of mouse and keyboard activity. Depending on the tasks participants were performing and how they performed those tasks, participants may or may not be prompted to complete the ESQ. For example, a task such as writing a document in a word processing program might require minimal use of the mouse (which generates a large volume of events) and greater use of the keyboard (generating comparatively fewer events than the mouse). Participants performing a PDF reading task would likely generate fewer mouse/keyboard events than someone else performing a photo editing task.

6.1.1.6 Individual Variation

There are individual variations in how a similar task is performed and these can affect how often participants were prompted to answer the questionnaire. For example, in the same word processing task, one person may use the mouse to select text and click the bold button, while another person may use a keyboard shortcut to select a word's text and Ctl-B to make the text boldface. The former case results in more mouse/keyboard events than the latter, and based on our software implementation, might also result in more questionnaire requests for the more mouse-intensive variation.

6.1.1.7 Awareness of Requests

Making participants aware of all ESQ requests could increase the number of completed ESQs. There is evidence from our results that some participants were unaware they were being requested to answer the ESQ. Research on notification interfaces [30], and on the peripheral perception of notification icons [8] can inform the design of better ESQ request notifications. Based on previous work, a number of strategies are possible for increasing awareness and we discuss these in the Future Work section of this chapter.

6.1.1.8 Skipping Requests

For many participants, a large proportion of ESQ requests were skipped; reducing the number of skipped questionnaires could increase the number of completed ESQs substantially. We did not ask participants why they skipped the questionnaire, but can presume a number of reasons. First, the user may have been performing a task they did not want to share with the researchers. For example, their keyboard activity may have contained sensitive (e.g., password) or personal (e.g., instant message to loved one) information. We designed the skip feature specifically to address these types of situations. Second, the questionnaire may have appeared at an inopportune time (e.g., a few minutes before lecture). We would like to support the ability to skip these questionnaires, but perhaps incentivize users enough to alter their perceived threshold of an inopportune time. Finally, users might have skipped the ESQ because they could not be bothered to fill it out. This third situation is the one that we would like to address. One way to decrease the number of skipped questionnaires is to provide sufficient incentive for participants to want to complete them. Research on motivating participation in ESM [36] could help us improve participation. We discuss later in the Lessons Learned section of this chapter.

Quantity of ESQs alone does not provide us with good data for predictive modeling. We also require data of sufficient quality, and we discuss that in our next subsection.

6.1.2 ESQ Data Quality

The quality of data is as important as the quantity of data for predictive modeling purposes. High ESQ data quality has two components. First, the data must provide an accurate assessment

of the ground truth emotional state of a participant. Second, the data must provide sufficient variation in responses, i.e., a balanced number of classes, to train a predictive model.

6.1.2.1 Ground Truth

The ground truth of participants' emotional states is essential for building accurate predictive models. Our method of assessing ground truth was using our ESQ to allow participants to report their level of emotional states on the 5-point Likert scale. In a survey of ESM studies, Csikszentmihalyi and Larson [18] found that the ESM results of internal state ratings to be similar to other techniques, thus we regard our participants' ratings to be as accurate as is possible for a single technique.

6.1.2.2 Balanced Ratings

Balanced ratings from participants are also important for predictive modeling. The problem of class skew is evident in the predictive models reported in Chapter 5. There are two potential sources of this. First, some participants provided a large number of questionnaire responses, but did not vary their responses. This indicates they may have been participating for the financial incentive rather than trying to provide good research data. Second, participants may not experience varying levels of some emotional states. We partially addressed the lack of rating variance by rescaling the 5-point scale to a 3-point scale, resulting in six emotional states with a minority class greater than 160 (an instance-feature ratio of 10) compared to none using the 5-point scale. This allowed us to perform predictive modeling on under-sampled data sets resulting in data sets with balanced class instance counts.

Despite the six balanced emotional states, our models still performed poorly. Next, we discuss the performance of our predictive models.

6.1.3 Predictive Modeling

Overall, the predictive models did not perform well. When models were created using the original, unmodified data set, six had prediction rates greater than sixty percent. However, none

of the Kappa values were greater than 0.30, indicating these models perform only slightly better than a random classifier. There was evidence of class skew with these models and when the data sets were adjusted for class skew with an instance-feature ratio of ten or higher, we saw prediction rates slightly better than chance with Kappa values below 0.20. Despite this, we are optimistic that mouse dynamics can be used to predict varying levels of emotional state. We have identified three reasons for optimism.

6.1.3.1 Feature Selection Can Be Improved

Our approach for selecting features could be improved. The features we used may have been a factor in the poorly performing models, but there are options for selecting potentially better features. We reduced the number of features from 38 to 16 using PCA to account for 95% of the variation in the data. We did not attempt other feature selection or reduction techniques. Machine learning algorithms behave differently with different types of data sets and a trial-and-error process of selecting features is often helpful for improving classifier performance [79] and furthermore, decision trees algorithms like C4.5 have been shown to perform worse when unnecessary attributes are used [79]. In our study, we were more concerned with building the software system and evaluating our approach rather than building the best models possible, so we did not spend much effort in trial-and-error modeling.

However, we did a simple test using a technique described by Witten & Frank [79] that shows how other feature selection techniques can work better. We chose the three highest level attributes used in the decision tree algorithm for classifying Happiness ratings – we used the normalized features, not the PCA generated features and the model was created on the original data set. Using these attributes on an under-sampled data set, we found the prediction rate was 57% with a Kappa value of 0.36, better than the 45% and 0.18 Kappa of the predictive model using an under-sampled dataset with PCA features. This example uses just one technique of many for selecting features and illustrates that a trial-and-error process for selecting features on our data can yield better classifier results.

As well as improving the selection of existing features, we can also derive new features from the raw mouse data we collected. Other studies have performed predictive modeling on participants' computers and reported results back to the experimenters [41], but this requires features be known when the software is deployed. Our approach differs because we collect raw data from participants' computers, allowing very flexible analysis of the data -- new features can be derived from the data. For example, identification of simple gestures like circling and the arc of the mouse cursor's line to a target may be important features and we have the data to calculate these.

6.1.3.2 Better Machine Learning Algorithms May Exist

The selection of the C4.5 algorithm was based on results in an early pilot study. Perhaps other algorithms would provide better results, but we did not evaluate other algorithms using our field study data. Further investigation into other machine learning algorithms should be done on the field study data to ascertain whether other algorithms provide better results. Although the algorithm or the researchers still must handle missing data, many options other than decision trees (e.g., support vector machines, neural networks) exist.

6.1.3.3 Intra-Participant Modeling Was Not Used

The volume of data collected did not allow us to perform intra-participant modeling. This is unfortunate because modeling individual participants may work better. Some studies have detected state differences in users by relying on intra-participant analysis [41]. Individual variations in mouse dynamics makes comparing across individuals difficult, which is why we relied on normalization for calculating values, so that we could compare across participants. Our normalization calculation does not take into account statistical variation in the values and perhaps normalization techniques that standardize the values would provide a better cross-participant comparison. However, if we had enough data to model individual participants, we could use non-normalized values, eliminating the need for normalization. Collecting data on active participants for a long study duration would allow us to investigate intra-participant modeling.

6.1.4 **Data Collection**

The data collection part of our approach worked well. The only problem we encountered during our field study was that two participants could not run our software because of a problem with the Client Recording Software not registering that participants had run the software previously. This caused the software to ask participants to fill out the demographic questionnaire every time the software started. This was a minor problem and did not affect our data collection. We did not fix the issue because it affected a small number of users, but will address it in future implementations of the Client Recording Software.

We envision enabling other modalities of predicting emotion such as facial expression recognition which has a proven history of predicting emotional states [7,26,65,81]. This is possible with the software system we designed and implemented by creating a new probe type. Combining affect recognition modalities provides more accurate predictions of user emotion [66]. Predictive models for mouse dynamics, keyboard dynamics, and facial expressions could be used to triangulate the emotional states users are experiencing.

Our software system already supports mouse and keyboard data, and could also handle several types of images and video. The software is capable of handling much more data than is currently being sent. During the field study, the system received, on average, just over one megabyte of data from participants' computers for each ESQ sample period and our pilot study indicated that up to twenty megabytes can be reliably delivered. Many computers, especially laptops, have web cameras that could be used by our software to capture images or video. The file sizes generated by these devices are within the capabilities of the current twenty megabyte limit of our system – a typical three megapixel image in JPEG format is approximately one megabyte, and one minute of video (640x480 pixel resolution and 18 frames per second using H.264/MPEG-4 encoding) is 6.4 megabytes. Also, as mentioned in the previous chapter, our system's bottleneck for receiving data was caused by the use of web services, specifically XML. Changing the technology to JSON would improve the throughput beyond twenty megabytes. Furthermore, our system could handle more than twenty megabytes by using more powerful hardware and changing the communication protocol used between the Client Recording Software and Data

Collection Web Service. The failures observed when messages were larger than twenty megabytes were memory related and our server only had 384 megabytes of RAM allocated to it. Increasing the RAM memory allocation would result in our system being able to handle larger packages of data from the Client Recording Software. Web services was used to send data to the server and the underlying protocol, SOAP, has greater memory requirements than other protocols such as JSON. Changing to a protocol requiring less memory would increase the throughput capacity of the system.

6.2 Lessons Learned

In this section, we discuss the lessons learned during the course of this research. First, we discuss the importance of knowing the number of minority instances needed for predictive modeling. Second, we describe how motivating participants through incentives can improve the completion rate of questionnaires. Finally, we discuss the importance of altering questionnaire frequency to increase ESQ completion rates.

6.2.1 Have a Goal for the Number of Minority Class Instances

Knowing the number of instances that need to be collected would allow researchers to be more proactive in altering participant behaviour to improve response rates and/or extending a study to improve the quantity and quality of data. The goal number of instances should be decided based on the number of minority class instances required to provide enough data for predictive modeling because this will allow the creation of predictive models based on balanced data, thus eliminating the class skew problem. It is also important to know if inter- or intra-participant modeling will be performed. If building intra-participant predictive models is the goal, then the goal applies to individual participant's data, not the all participants' data.

In many cases, it will be difficult to have a goal number of minority class instances before the study begins. Feature selection techniques, such as PCA, require data to determine features. Recall that PCA will select features to represent a certain percentage of variation in the data, 95% in our case. This means, the number of required features is unknown until the data is

collected. The number of features affects the number of instances needed to represent the problem space sufficiently for classification – each dimension (feature) requires a certain number of samples to train a model. This is why the heuristic instance-feature ratio of ten exists [40].

Our approach does not allow researchers to know the number of features before the study, but it does allow researchers to analyze participants' data while the study is in-progress. The data collection design of the software system means participant data is available immediately after they have completed the ESQ and predictive modeling can be performed on an on-going basis. This would allow researchers to continuously evaluate the number of features needed to perform predictive modeling and adjust the goal number of instances.

6.2.2 **Motivate Participants**

Motivating participants to complete more questionnaires may increase the quantity and the quality of data by reducing the number of skipped questionnaires and encouraging participants to answer questionnaires when they are in minority class states. We did nothing to motivate our participants based on the data they were providing us.

One problem with our approach is we did not provide feedback to participants on how they were contributing to the field study. Recall that each week, participants who completed at least fifteen ESQs had their names added to a draw to win one of three $100 gift certificates. We provided an incentive for participants to provide a high quantity of questionnaire responses, but we did not provide them incentive for the other aspect we wanted in the data, quality, and at least one participant took advantage of this by only slightly varying their responses. Furthermore, we did not provide participants with any feedback about the quantity or the quality of the data they were providing. Participants were not even notified whether they were or were not put into the weekly draw.

Providing feedback is a low cost and simple way to motivate participants to provide greater quantity and higher quality data. Hsieh et al. [36] found in a study of ESM with feedback that participants who could view a report of their responses had significantly higher response rates

than participants who could not. There are ways of providing feedback as well that may improve response rates. We can think of a few:

- a static weekly email reminding participants of their commitment to the study and the importance of their continued participation.
- a weekly email indicating the number of questionnaires completed for the previous week and/or the entire study
- the number of skipped questionnaires with a reminder that it is valuable for us to collect as many questionnaires as possible
- after the questionnaire is completed, show participants how they have been responding

We can also think of some types of feedback that might not work well because they could affect the ground truth of the data. For example, sending information to participants regarding how close they are to the goals for minority classes may cause participants to misrepresent their emotional states so they can complete the study sooner. Another example is sending participants information comparing them to other participants on measures such as the number of ESQs completed. Such information may result in participants answering questionnaires as quickly as possible without regard for their actual emotional state.

6.2.3 **Alter Questionnaire Frequency**

A shorter interval between ESQ requests would result in more completed questionnaires, assuming the response rate was the same in both cases. This would be a simple way to increase the number of questionnaires. The system could be modified to allow researchers to dynamically increase or decrease the interval between questionnaires after the software is deployed to participants' computers.

6.2.4 **Adaptive Techniques**

One of the main drawbacks to ESM is that the balance of responses is generally uncontrolled. In our study, this manifested in class skew from unbalanced distributions. A solution for improving the balance across ESQ categories is to use an adaptive technique to intelligently prompt the user

with the ESQ. If predictive models could be generated from a small amount of initial data, the software could monitor the user's emotional state using these preliminary models. When the software detected a low-frequency state (e.g., high anger), it could prompt the user with the ESQ to try to better balance the distribution between the classes. Initial preliminary models with some degree of accuracy are needed; however, these could be generated part way through a study or could be generated in one study to be used in a subsequent study. In our case, the keystroke data gathered from our field study was successfully used to generate models of a number of emotional states [25]. We could use the results of the keystroke models to adapt our ESQ prompting for a new study examining mouse dynamics alone.

6.3 Future Work

In this section, we discuss three future areas of work that we plan to undertake. First, we describe how we can perform further analysis of the existing data already collected. Next, describe a study analyzing feedback provided while conducting ESM studies. Finally, we describe a plan for using attentional draw to reduce ignored ESQ requests.

6.3.1 Further Explore Collected Data

Our predictive modeling of the field study data was performed to validate our overall approach for collecting affective field study data. As mentioned earlier in this chapter, we put less effort than we could have into the trial-and-error process that is often required in predictive modeling. There are a number of areas we can take to improve the performance of our predictive models.

First, we can derive new features from the data. We collected data indicating the applications that were open and the application with focus at 10-second intervals, but we did not use these data in our modeling. One problem we encountered using these data was that participants used different software for the same class of use. For example, Internet Explorer and Firefox were two different web browsers used. This can be solved by aggregating these different applications into application classes, e.g., web browsers, word processors, text editors, and games. We could improve predictive modeling is by using different sampling time periods. We arbitrarily chose

5-minutes as the sample time period for calculating the mouse summary values used as features. Our reasoning for this time period is that it may bridge the time gap between emotions and their longer-lived counterpart, moods. Summarizing the mouse data using different time periods such as thirty seconds, one minute and two minutes may give different predictive modeling results. These new features could provide quite different data for machine learning algorithms and we think will be worthwhile to take advantage of the existing data we have. They are also not that difficult to derive from the data.

Second, we can perform different feature-selection techniques to reduce the dimensionality of the problem space. In the Summary of Findings section of this chapter, we gave an example of how feature selection can be improved by using a small number of the top level features of a decision tree. We will explore the use of other feature selection techniques such as forward selection and backward elimination [79]. In forward selection, a subset of features is generated by starting with an empty subset and iterating through all features. In each iteration, a predictive model is evaluated using the features in the subset and the feature for that iteration. After all iterations are completed, if there was a feature that when combined with the subset provided better performance than the subset, it is selected be added to the subset and removed from the set of all features. The process is repeated until no more features can be found that improve predictive model performance. Backward elimination is similar except the starting feature subset is all the features and the feature that causes the subset to perform the worst when removed, is eliminated from the feature subset. Feature selection, including these techniques, is a time intensive operation.

Third, we only used the C4.5 machine learning algorithm for our evaluation. Other algorithms will be explored such as neural networks.

6.3.2 **ESM with Feedback**

Feedback has been found to improve ESM response rates [36]. We would like to explore a variety of feedback techniques with the aim of determining which increase response rates the most and incorporating those into the system. There are two aspects we are interested in. First,

we want to know which information results in higher response rates. The information displayed to participants could be total number of responses, rating counts for each emotional state, or comparing the participant to others using a metric that is neutral towards computer usage time (i.e., proportion of questionnaires completed to questionnaires requested). Second, we want to know whether visualizations are better than just textual representations.

6.3.3 **Attentional Draw to Reduce Ignored Requests**

Increasing levels of attentional draw [30,58] can be used to attract users' attention to fill out a questionnaire. Our implementation for prompting users used a simplistic approach that used only one attentional draw technique: the taskbar balloon. We could use a variety of techniques with increasing attentional draw so that participants who might ignore the questionnaire initially will be required to explicitly skip it. For example, we could alter the taskbar icon to a flashing image (the least intrusive technique), followed by showing a taskbar balloon, and finally show a pop-up window. All options would still allow participants to skip the questionnaire.

7 CONCLUSION

7.1 Summary

In this thesis, we have described and evaluated an ESM software system for creating predictive models of users' emotional states. The system consisted of five parts. The Client Recording Software captured ESQ (experience sampling questionnaire) responses and data from the mouse and keyboard that was used to generate predictive modeling features. This software was installed on participant computers and communicated with the second part, the Data Collection Web Service, to deliver experiment data. The Field Study Web Application was used to view and retrieve experimental data using a web browser. The Daily Reports Web Application allowed us to view the progress of individual participants in terms of the number of questionnaires answered. Finally, Data Summarization Scripts were used to generate predictive modeling features from the experimental data.

7.1.1 Research Questions

The ESM software system was evaluated using a field study to answer five research questions and address two main goals.

Our first goal was to *validate our approach for detecting participants' emotions states*. We asked four central questions:

1. **Can we determine ground truth?**

   We found that we can determine ground truth by using the ESQ. 1,555 questionnaires were completed by twenty-six participants over a two month study period and these were successfully delivered to our Data Collection Web Service.

2. **Can we collect data for predictive modeling?**

   We found the system did collect data for predictive modeling. The system collected 1,555 instances for predictive modeling that contained both mouse and keyboard data. We found that the throughput of the system could handle enough data to collect video data for facial expression recognition.

3. **Can we build predictive models from the data?**

   We were able to build predictive models from the data collected. The Data Summarization Scripts generated thirty-eight mouse features that were used for predictive modeling.

4. **Can we reduce the logistical difficulty in carrying out these processes?**

   Our software system automated many of the difficult aspects of performing an ESM affective computing field study. Computerized ESM allowed us to prompt participants to complete questionnaires only when there was enough mouse and keyboard data to perform predictive modeling. User data and ground truth were matched using our delivery process. Data collection was automatic and therefore, did not rely on participants sending log files. The Data Retrieval Web Application allowed us to download participants' data any time. Summarization of data was handled through scripts and allowed re-generation of features at any time. The combination of these parts of our system automated the process of gathering and labeling field data and allowed us to easily generate predictive models and iterate on the modeling process.

Our second goal was to *determine whether any emotional states could be detected in participants*. We asked one question:

5. **Can we detect emotional states using our approach?**

   We were not able to predict emotional states from the mouse data, but this does not invalidate the success of our approach. A related study using the keyboard data

collected during the field study – using our software system – was able to predict six emotional states with prediction rates between 77 and 88%, and is awaiting publication [25]. Furthermore, we identified three reasons to be optimistic that emotional states could still be predicted from our data: feature selection can be improved, better machine learning algorithms may exist, and intra-participant modeling was not used.

7.1.2 Roadblocks Have Been Addressed

In our problem statement, we identified five roadblocks for performing affective computing field studies: ground truth, data collection, predictive modeling, uncontrolled factors, and unknown frequency and duration of emotions. In our solution, we sought to eliminate the first three roadblocks. We created a software system that was able to determine ground truth, collect data, and perform predictive modeling on data gathered from a field study. In addition, we present ideas in our discussion for experimentally dealing with the final roadblock of unknown frequency and duration of emotion. The fourth roadblock – that there are many uncontrolled factors influencing a user's emotion – will still be true in any adaptive emotionally-aware system that is developed. Future research will need to create systems that can not only identify a user's emotional state, but can also identify the cause.

7.2 Contributions

7.2.1 Main Contribution

Our main contribution is that we have designed, implemented, and tested a system for conducting experience sampling studies for affective computing. Our software solution was able to gather users' mouse and keyboard data from their computers. The system implemented computerized ESM and through the ESQ, was able to determine participants' ground truth emotional states. The delivery design enabled emotional state and user data to be delivered together over the internet to a central collection server, and allowed researchers to monitor the

user study remotely. And finally, our summarization process prepared the emotional state and user data for the predictive modeling process.

7.2.2 Secondary Contributions

We have secondary contributions as well based on the evaluation of our software system. We have identified a number of areas that can improve future field studies using computerized ESM such as knowing the number of minority class instances needed, continuously modeling during the data collection phase to learn what the minority class instance count should be, motivating participants, altering questionnaire frequency, and using adaptive techniques to intelligently prompt participants to complete the questionnaire with the goal of increasing the number of minority class instances.

7.3 Conclusion

The main contribution described in this thesis, our software system, solves three of the five points in our problem statement – it records ground truth, collects ground truth and predictive modeling data, and generates features for predictive modeling. Our approach uses the ESM, providing ecological validity and making realistic emotion data available to researchers, which is a necessary step towards finding real-world solutions for determining user emotional states in users' every day computing environment. Once emotional states are detectable in people's everyday computer environments, software developers can begin incorporating emotional state awareness into their products, and we will be closer to the goal of creating more emotionally-aware and expressive software.

LIST OF REFERENCES

1. affect - Wiktionary. http://en.wiktionary.org/wiki/affect.
2. Dimension reduction - Wikipedia, the free encyclopedia. http://en.wikipedia.org/wiki/Dimension_reduction.
3. Principal component analysis - Wikipedia, the free encyclopedia. http://en.wikipedia.org/wiki/Principal_components.
4. iLab Cookbook - Windows Hooker. http://grouplab.cpsc.ucalgary.ca/cookbook/index.php/Toolkits/WindowsHooker.
5. Health informatics - Point-of-care medical device communication - Part 10201: Domain information model. *ISO/IEEE 11073-10201:2004(E)*, 2004.
6. Alexander, J., Cockburn, A., and Lobb, R. AppMonitor: a tool for recording user actions in unmodified Windows applications. *Behavior Research Methods 40*, 2 (2008), 413-421.
7. Bailenson, J.N., Pontikakis, E.D., Mauss, I.B., et al. Real-time classification of evoked emotions using facial feature tracking and physiological responses. *International Journal of Human-Computer Studies 66*, 5 (2008), 303-317.
8. Bartram, L., Ware, C., and Calvert, T. Moticons:: detection, distraction and task. *International Journal of Human-Computer Studies 58*, 5 (2003), 515-545.
9. Batliner, A., Fischer, K., Huber, R., Spilker, J., and Nöth, E. Desperately seeking emotions or: Actors, wizards, and human beings. *Proceedings of the International Speech Communication Association Workshop on Speech and Emotion*, (2000), 195-200.
10. Bradley, M.M. and Lang, P.J. The International Affective Picture System (IAPS) in the Study of Emotion and Attention. In J.A. Coan and J.J.B. Allen, eds., *Handbook of Emotion Elicitation and Assessment*. Oxford University Press, New York, NY, USA, 29-46.
11. Bradley, M.M. and Lang, P.J. Measuring emotion: The self-assessment manikin and the semantic differential. *Journal of Behavior Therapy and Experimental Psychiatry 25*, 1 (1994), 49-59.
12. Bradley, M.M. and Lang, P.J. *International Affective Digitized Sounds (IADS): Stimuli, instruction manual and affective ratings*. University of Florida, Gainesville, FL, USA, 1999.
13. Bradley, M.M. and Lang, P.J. *Affective Norms for English Words (ANEW): Instruction manual and affective ratings*. University of Florida, Gainesville, FL, USA, 1999.
14. Bradley, M.M. and Lang, P.J. *Affective Norms for English Text (ANET): Affective ratings of text and instruction manual*. University of Florida, Gainesville, FL, USA, 2007.
15. Chawla, N.V., Bowyer, K.W., Hall, L.O., and Kegelmeyer, W.P. SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research 16*, 1 (2002), 321-357.
16. Cohen, I., Garg, A., and Huang, T.S. Emotion Recognition from Facial Expressions using Multilevel HMM. *NIPS Workshop on Affective Computing*, (2000).
17. Crockford, D. RFC 4627: The application/json Media Type for JavaScript Object Notation (JSON). http://www.ietf.org/rfc/rfc4627.txt.

18. Csikszentmihalyi, M. and Larson, R. Validity and reliability of the Experience-Sampling Method. *The Journal of Nervous and Mental Disease 175*, 9 (1987), 526-536.
19. Devillers, L., Vidrascu, L., and Lamel, L. Challenges in real-life emotion annotation and machine learning based detection. *Neural Networks 18*, 4 (2005), 407-422.
20. Dornaika, F. and Raducanu, B. Inferring facial expressions from videos: Tool and application. *Signal Processing: Image Communication 22*, 9 (2007), 769-784.
21. Ekman, P., Levenson, R., and Friesen, W. Autonomic nervous system activity distinguishes among emotions. *Science 221*, 4616 (1983), 1208-1210.
22. Ekman, P. Universals and cultural differences in facial expressions of emotion. *Nebraska Symposium on Motivation (1971)*, University of Nebraska Press (1972), 207-282.
23. Ekman, P. An argument for basic emotions. *Cognition & Emotion 6*, 3 (1992), 169-200.
24. Ekman, P., Friesen, W.V., and Ancoli, S. Facial signs of emotional experience. *Journal of Personality and Social Psychology 39*, 6 (1980), 1125-1134.
25. Epp, C.C., Lippold, M.T., and Mandryk, R.L. Identifying Emotional States using Keystroke Dynamics. *[ACCEPTED] Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, (2011).
26. Essa, I. and Pentland, A. Facial expression recognition using a dynamic model and motion energy. *Computer Vision, 1995. Proceedings., Fifth International Conference on*, (1995), 360-367.
27. Fielding, R.T. Architectural Styles and the Design of Network-based Software Architectures. 2000.
28. Fowler, M. *Analysis patterns: reusable objects models*. Addison-Wesley Longman Publishing Co., Inc., 1997.
29. Froehlich, J., Chen, M.Y., Consolvo, S., Harrison, B., and Landay, J.A. MyExperience. *Proceedings of the 5th International Conference on Mobile Systems, Applications and Services - MobiSys '07*, (2007), 57-70.
30. Gluck, J., Bunt, A., and McGrenere, J. Matching attentional draw with utility in interruption. *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '07*, (2007), 41-50.
31. Gomez, P., Zimmermann, P., Guttormsen-Schär, S., and Danuser, B. Valence Lasts Longer than Arousal. *Journal of Psychophysiology 23*, 1 (2009), 7-17.
32. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I.H. The WEKA data mining software: an update. *SIGKDD Explor. Newsl. 11*, 1 (2009), 10-18.
33. Hancock, J.T., Gee, K., Ciaccio, K., and Lin, J.M.H. I'm sad you're sad: emotional contagion in CMC. *Proceedings of the ACM 2008 Conference on Computer Supported Cooperative Work*, ACM New York, NY, USA (2008), 295-298.
34. Hassenzahl, M. Hedonic, emotional, and experiential perspectives on product quality. *Encyclopedia of Human Computer Interaction*, (2006), 266-272.
35. Hofmann, C., Weigand, C., and Bernhard, J. Wireless medical sensor network with ZigBee™. *Proceedings of the 6th WSEAS International Conference on Applied Informatics and Communications*, World Scientific and Engineering Academy and Society (WSEAS) (2006), 116-119.
36. Hsieh, G., Li, I., Dey, A., Forlizzi, J., and Hudson, S.E. Using visualizations to increase compliance in experience sampling. *Proceedings of the 10th International Conference on*

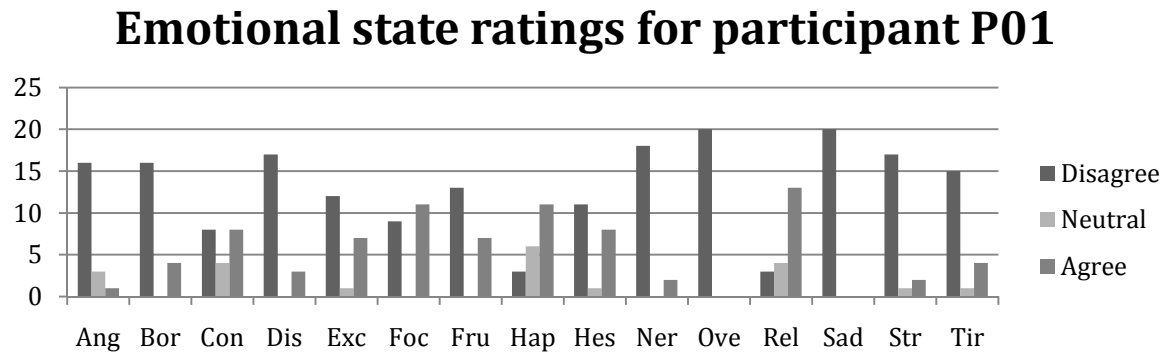
Ubiquitous Computing, (2008), 164-167.
37. Hurst, A., Hudson, S.E., and Mankoff, J. Dynamic detection of novice vs. skilled use without a task model. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM Press (2007), 271-280.
38. Hurst, A., Hudson, S.E., and Mankoff, J. Automatically detecting pointing performance. *Proceedings of the 13th International Conference on Intelligent User Interfaces*, ACM (2008), 11-19.
39. Isomursu, M., Tähti, M., Väinämö, S., and Kuutti, K. Experimental evaluation of five methods for collecting emotions in field settings with mobile applications. *International Journal of Human-Computer Studies 65*, 4 (2007), 404-418.
40. Jain, A. and Chandrasekaran, B. Dimensionality and sample size considerations in pattern recognition practice. In *Classification Pattern Recognition and Reduction of Dimensionality*. Elsevier, 1982, 835-855.
41. Kapoor, A. and Horvitz, E. Experience sampling for building predictive user models: a comparative study. *Proceedings of the 26th International Conference on Human Factors in Computing Systems*, ACM (2008), 657-666.
42. Kapoor, A., Burleson, W., and Picard, R.W. Automatic prediction of frustration. *International Journal of Human-Computer Studies 65*, 8 (2007), 724-736.
43. Karapanos, E., Zimmerman, J., Forlizzi, J., and Martens, J. User experience over time. *Proceedings of the 27th International Conference on Human Factors in Computing Systems - CHI '09*, (2009), 729-738.
44. Keates, S. and Trewin, S. Effect of age and Parkinson's disease on cursor positioning using a mouse. *Proceedings of the 7th International ACM SIGACCESS Conference on Computers and Accessibility - Assets '05*, (2005), 68-75.
45. Keinan, G. and Eilat-Greenberg, S. Can Stress be Measured by Handwriting Analysis? The Effectiveness of the Analytic Method. *Applied Psychology 42*, 2 (1993), 153-170.
46. Kirsch, D. *The Sentic Mouse: Developing a tool for Measuring Emotional Valence*. MIT Media Laboratory Perceptual Computing Section, 1997.
47. Kleinginna, P.R. and Kleinginna, A.M. A categorized list of emotion definitions, with suggestions for a consensual definition. *Motivation and Emotion 5*, 4 (1981), 345-379.
48. Kort, J., Vermeeren, A., and Fokker, J.E. Conceptualizing and Measuring User eXperience. *Towards a UX Manifesto*, , 57-64.
49. Larson, J.T., Berntson, G.G., Poehlmann, K.M., Ito, T.A., and Cacioppo, J.T. The Psychophysiology of Emotion. In *Handbook of Emotions*. The Guilford Press, New York, NY, USA, 2008, 180-195.
50. Larsson, S., Larsson, R., Zhang, Q., Cai, H., and Ake Oberg, P. Effects of psychophysiological stress on trapezius muscles blood flow and electromyography during static load. *European Journal of Applied Physiology and Occupational Physiology 71*, 6 (1995), 493-498.
51. Laursen, B., Jensen, B.R., Garde, A.H., and Jorgensen, A.H. Effect of mental and physical demands on muscular activity during the use of a computer mouse and a keyboard. *Scandinavian Journal of Work, Environment & Health 28*, 4 (2002), 215-221.
52. Law, E.L., Roto, V., Hassenzahl, M., Vermeeren, A.P., and Kort, J. Understanding, scoping and defining user experience: a survey approach. *Proceedings of the 27th International*

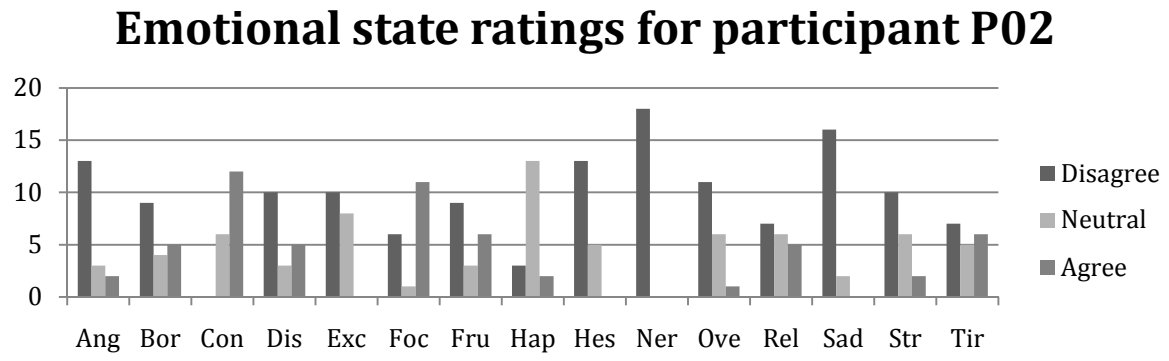
Conference on Human Factors in Computing Systems, ACM (2009), 719-728.
53. Leaning, M., Yates, C., Patterson, D., Ambroso, C., Collinson, P., and Kalli, S. A data model for intensive care. *International Journal of Clinical Monitoring and Computing 8*, 3 (1991), 213-224.
54. Lin, B., Lin, B., Chou, N., Chong, F., and Chen, S. RTWPMS: A Real-Time Wireless Physiological Monitoring System. *IEEE Transactions on Information Technology in Biomedicine 10*, 4 (2006), 647-656.
55. Lindquist, K.A. and Barrett, L.F. Emotional complexity. In M. Lewis, J.M. Haviland-Jones and L.F. Barrett, eds., *Handbook of Emotions*. The Guilford Press, New York, NY, USA, 2008, 513-530.
56. Mandryk, R.L. and Atkins, M.S. A fuzzy physiological approach for continuously modeling emotion during interaction with play technologies. *International Journal of Human-Computer Studies 65*, 4 (2007), 329-347.
57. Mandryk, R.L., Atkins, M.S., and Inkpen, K.M. A continuous and objective evaluation of emotional experience with interactive play environments. *Proceedings of the SIGCHI conference on Human Factors in computing systems*, ACM (2006), 1027-1036.
58. Masarakal, M. Improving Expertise-Sensitive Help Systems. 2010.
59. Mehrabian, A. and Russell, J.A. *An Approach to Environmental Psychology*. The MIT Press, Cambridge, Massachusetts, 1974.
60. Nacke, L.E. and Grimshaw, M. Player-game Interaction Through Affective Sound. In *Game Sound Technology and Player Interaction: Concepts and Developments*. IGI Global Publishing, Hershey, PA, USA, 2011, 264-285.
61. Napa Scollon, C., Prieto, C., and Diener, E. Experience Sampling: Promises and Pitfalls, Strength and Weaknesses. In *Assessing Well-Being*. 2009, 157-180.
62. Nelwan, S., van Dam, T., Klootwijk, P., and Meij, S. Ubiquitous mobile access to real-time patient monitoring data. *Computers in Cardiology, 2002*, (2002), 557-560.
63. Norgall, T., Schmidt, R., and von der Grün, T. Body area network -- a key infrastructure element for patient-centered telemedicine. *Studies in Health Technology and Informatics 108*, (2004), 142-148.
64. Obrist, M., Roto, V., and Väänänen-Vainio-Mattila, K. User experience evaluation. *Proceedings of the 27th International Conference Extended Abstracts on Human Factors in Computing Systems - CHI EA '09*, (2009), 2763.
65. Partala, T., Surakka, V., and Vanhala, T. Real-time estimation of emotional experiences from facial expressions. *Interacting with Computers 18*, 2 (2006), 208-226.
66. Picard, R.W. *Affective computing*. MIT Press, 1997.
67. Riseberg, J., Klein, J., Fernandez, R., and Picard, R.W. Frustrating the user on purpose: using biosignals in a pilot study to detect the user's emotional state. *CHI 98 Conference Summary on Human Factors in Computing Systems*, ACM (1998), 227-228.
68. Rottenberg, J., Ray, R.D., and Gross, J.J. Emotion Elicitation Using Films. In J.A. Coan and J.J.B. Allen, eds., *Handbook of Emotion Elicitation and Assessment*. Oxford University Press, New York, NY, USA, 2007, 9-28.
69. Russell, J.A. A circumplex model of affect. *Journal of Personality and Social Psychology 39*, 6 (1980), 1161-1178.
70. Scheirer, J., Fernandez, R., Klein, J., and Picard, R.W. Frustrating the user on purpose: a step

toward building an affective computer. *Interacting with Computers 14*, 2 (2002), 93-118.
71. Schlosberg, H. The description of facial expressions in terms of two dimensions. *Journal of Experimental Psychology 44*, 4 (1952), 229-237.
72. Shaw, M. and Garlan, D. *Software architecture: perspectives on an emerging discipline*. Prentice-Hall, Inc., 1996.
73. Silveira, M.H., Nunn, C., Lakhanpal, A., McDonagh, D., McPartland, R., and Burdett, A. Key Considerations and Experience Using the Ultra Low Power Sensium Platform in Body Sensor Networks. *Proceedings of the 2009 Sixth International Workshop on Wearable and Implantable Body Sensor Networks*, IEEE Computer Society (2009), 262-266.
74. Tellegen, A., Watson, D., and Clark, L.A. On the Dimensional and Hierarchical Structure of Affect. *Psychological Science 10*, 4 (1999), 297 -303.
75. Theodoridis, S. and Koutroumbas, K. *Pattern Recognition*. Academic Press, 2009.
76. Tomkins, S.S. *Affect, Imagery, Consciousness: The positive affects*. Springer Publishing Company, New York, NY, USA, 1962.
77. Varady, P. and Benyo, Z. An open architecture patient monitoring system using standard technologies. *IEEE Transactions on Information Technology in Biomedicine 6*, 1 (2002), 95-98.
78. Wheeler, L. and Reis, H.T. Self-Recording of Everyday Life Events: Origins, Types, and Uses. *Journal of personality 59*, 3 (1991), 339-354.
79. Witten, I.H. and Frank, E. *Data Mining: Practical Machine Learning Tools and Techniques.* Morgan Kaufmann, San Francisco, 2005.
80. Yoo, J., Cho, N., and Yoo, H. Analysis of body sensor network using human body as the channel. *Proceedings of the ICST 3rd international conference on Body area networks*, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering) (2008), 1-4.
81. Zhihong Zeng, Pantic, M., Roisman, G., and Huang, T. A Survey of Affect Recognition Methods: Audio, Visual, and Spontaneous Expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence 31*, 1 (2009), 39-58.
82. Zimmermann, P., Gomez, P., Danuer, B., and Schär, S.G. Extending usability: putting affect into the user-experience. *Proceedings of NordiCHI '06*, (2006), 27-32.

APPENDIX A EMOTIONAL STATE RATINGS BY PARTICIPANT

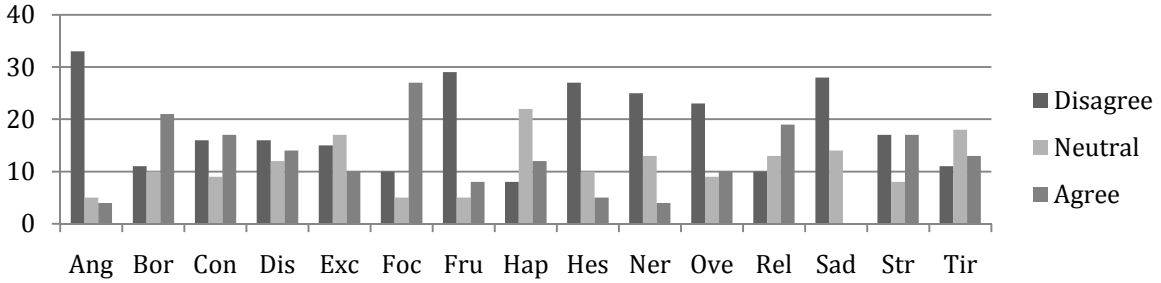
This appendix contains the 3-class emotional state ratings for all participants except P15 and P25 because they only participated for one day.



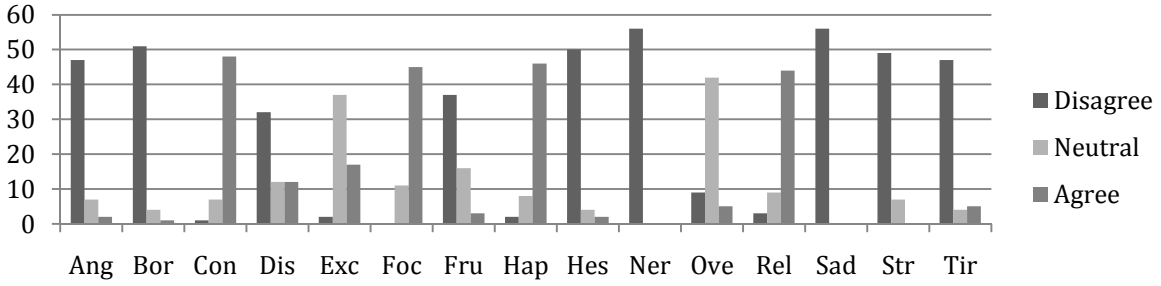


Emotional state ratings for participant P03

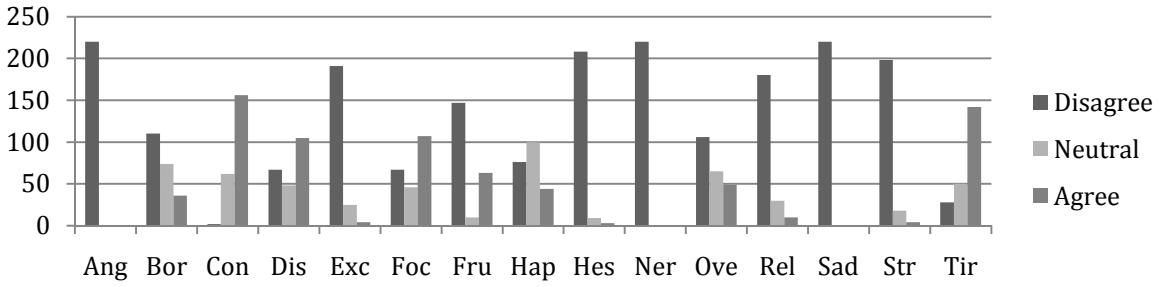
Emotional state ratings for participant P04



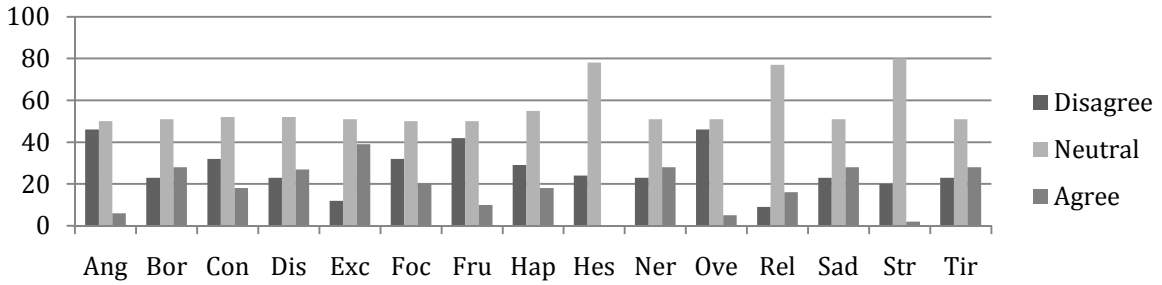
Emotional state ratings for participant P05



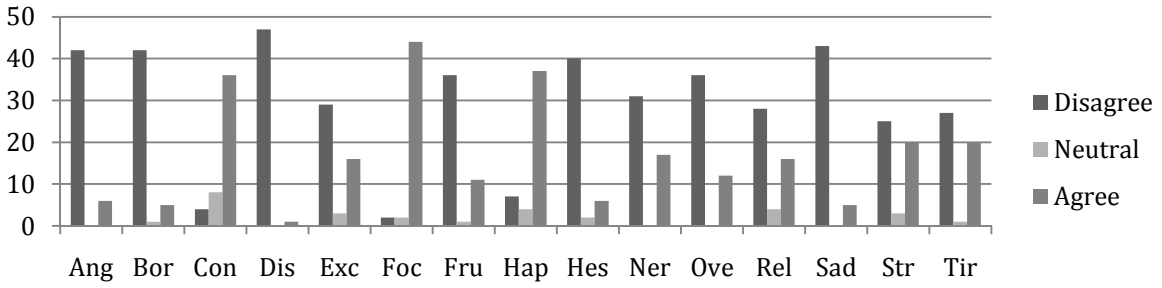
Emotional state ratings for participant P06

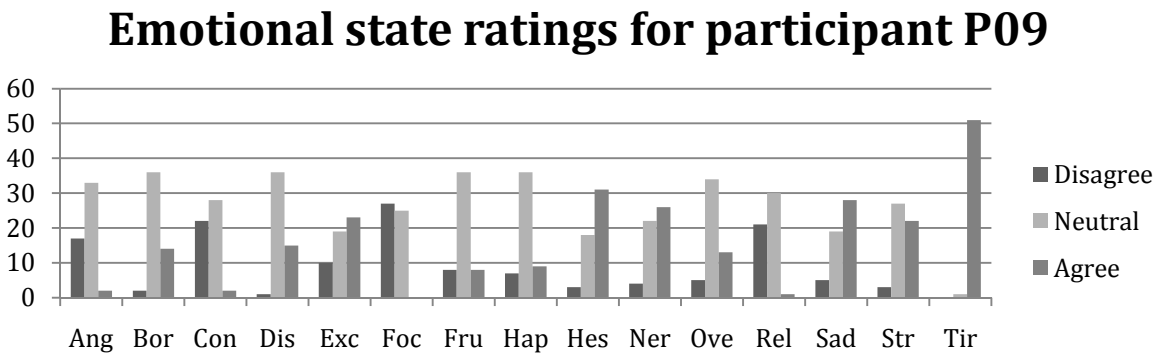


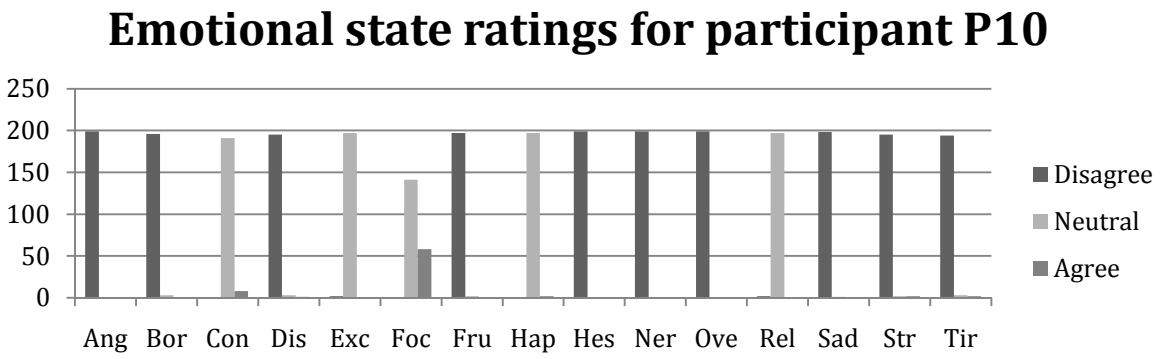
Emotional state ratings for participant P07

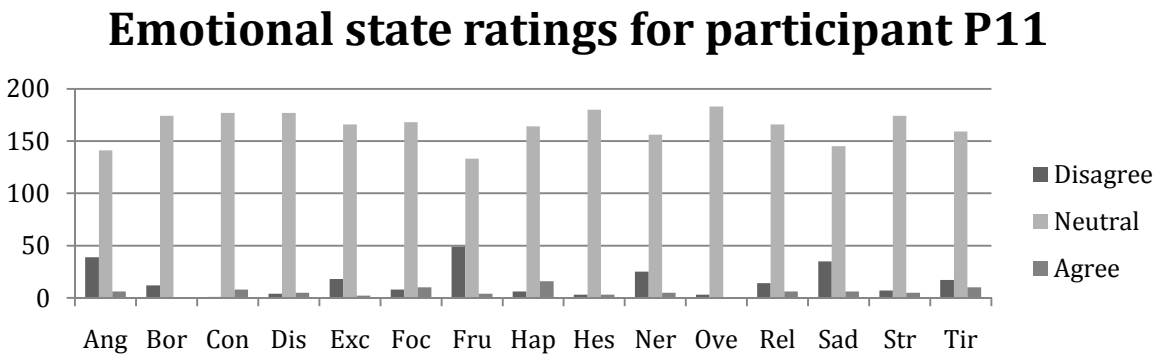


Emotional state ratings for participant P08



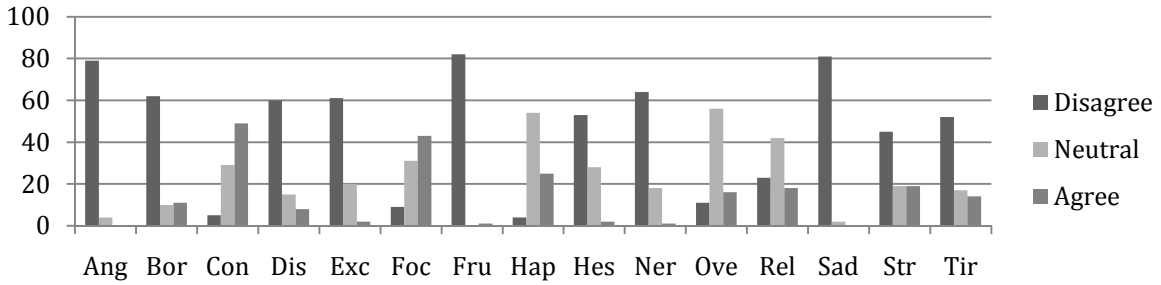

Emotional state ratings for participant P09
60
50
40
30
20
10
0
Ang Bor Con Dis Exc Foc Fru Hap Hes Ner Ove Rel Sad Str Tir
Disagree
Neutral
Agree


Emotional state ratings for participant P10
250
200
150
100
50
0
Ang Bor Con Dis Exc Foc Fru Hap Hes Ner Ove Rel Sad Str Tir
Disagree
Neutral
Agree

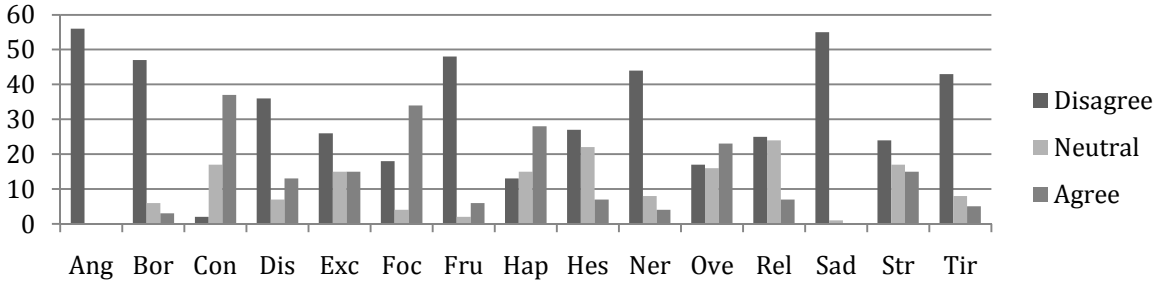

Emotional state ratings for participant P11
200
150
100
50
0
Ang Bor Con Dis Exc Foc Fru Hap Hes Ner Ove Rel Sad Str Tir
Disagree
Neutral
Agree

Emotional state ratings for participant P12

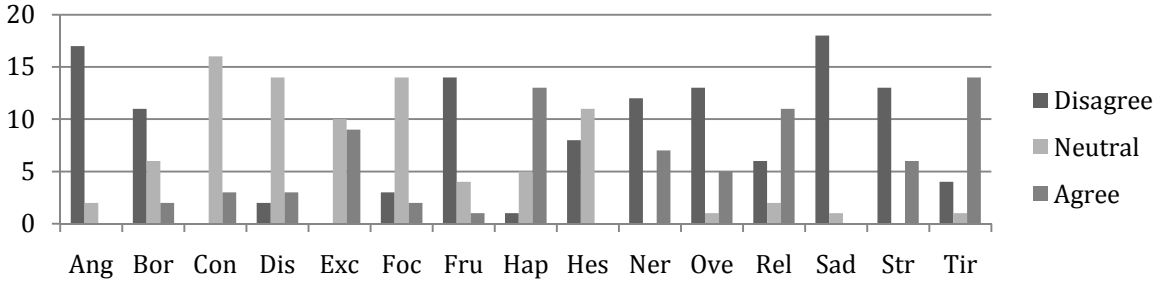
Emotional state ratings for participant P13



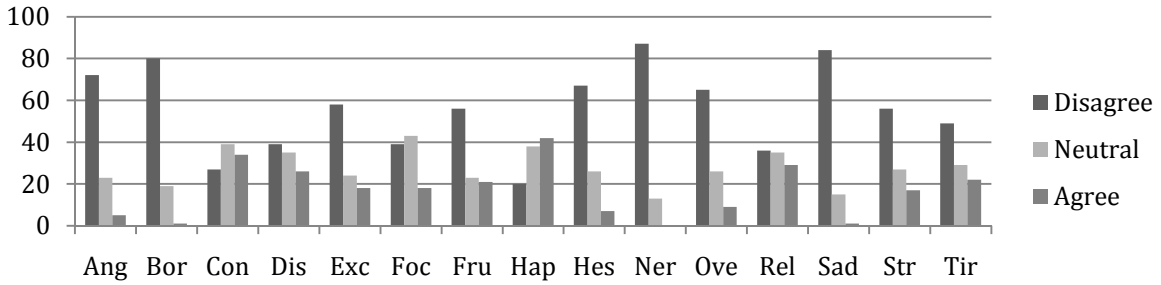
Emotional state ratings for participant P14



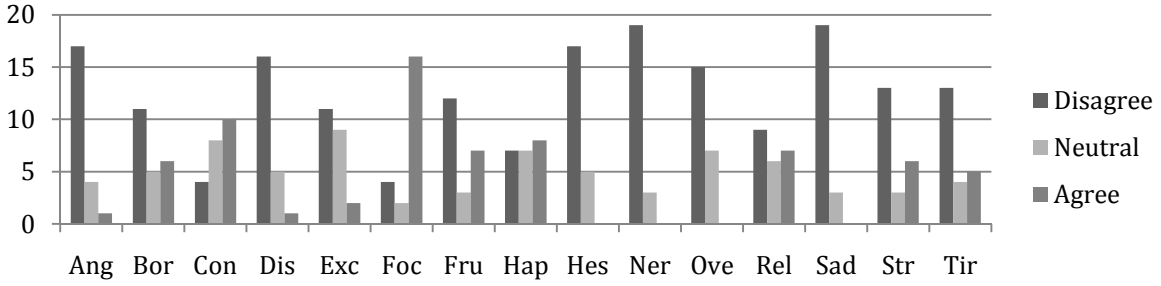
Emotional state ratings for participant P16



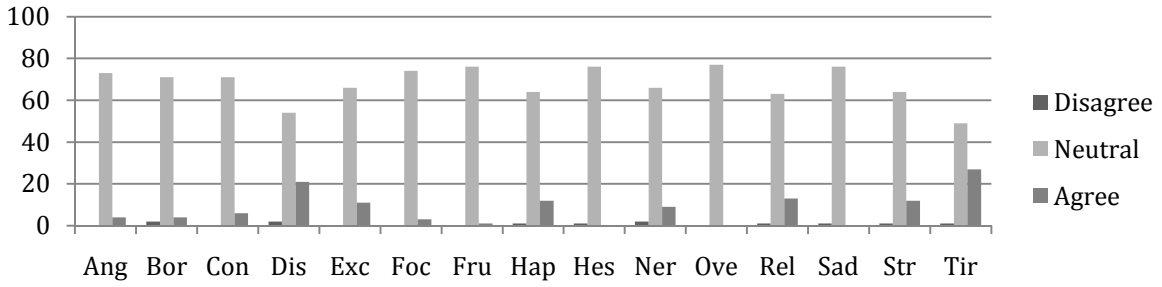
Emotional state ratings for participant P17



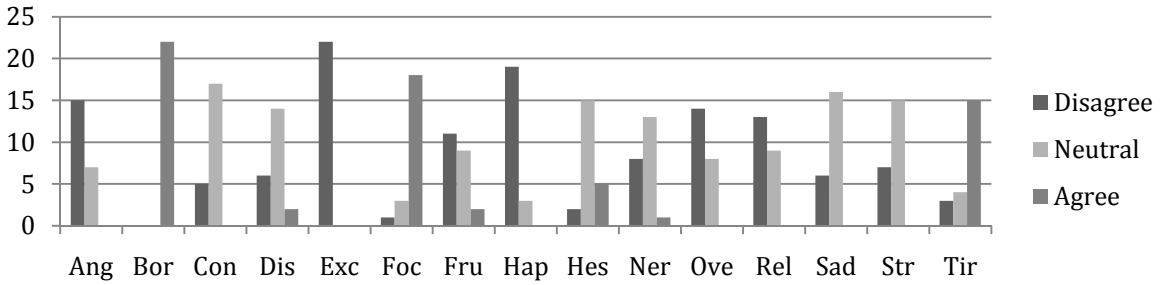
Emotional state ratings for participant P18



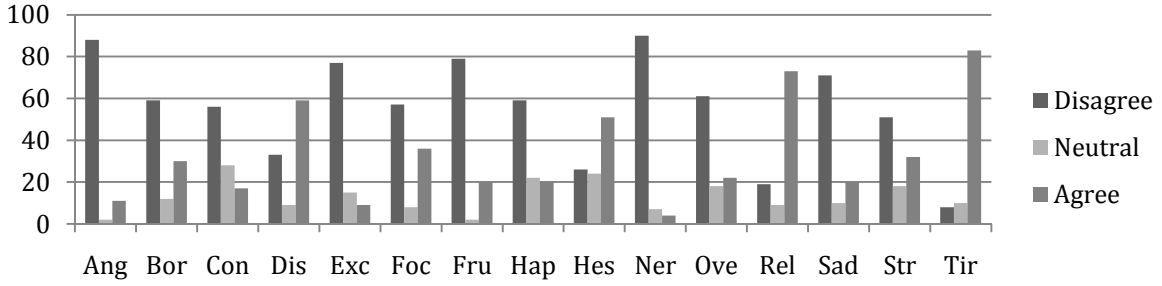
Emotional state ratings for participant P19



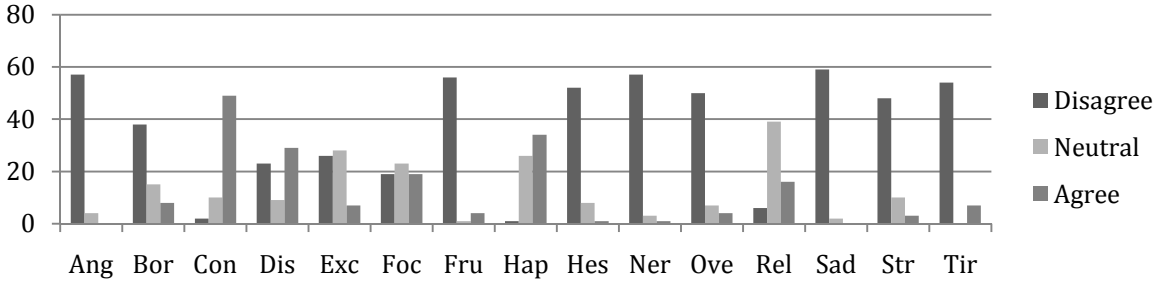
Emotional state ratings for participant P20



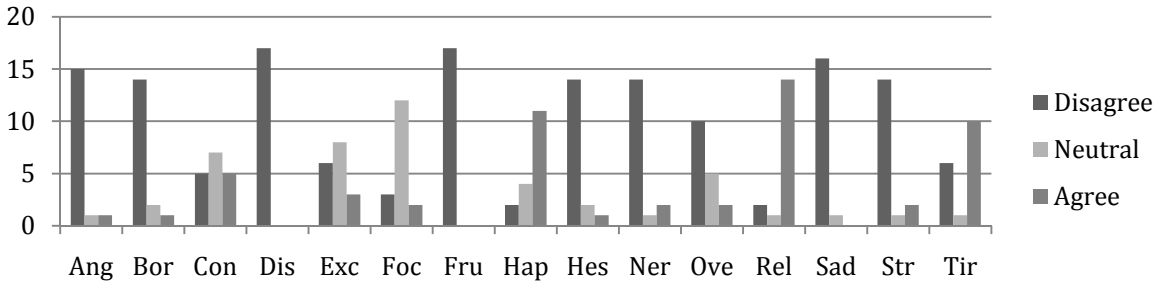
Emotional state ratings for participant P21



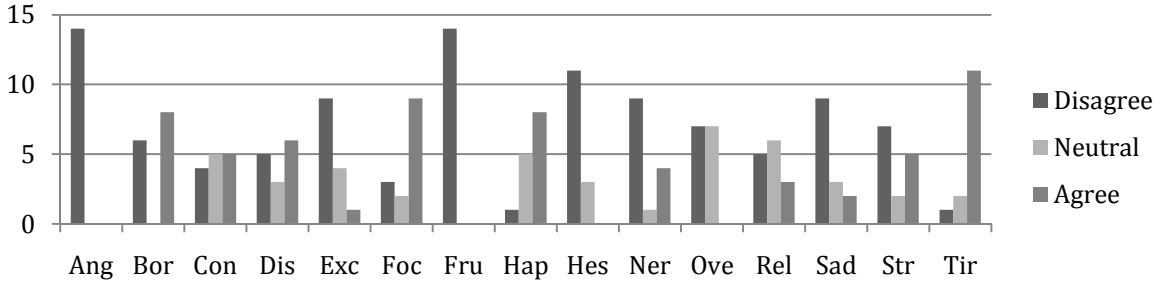
Emotional state ratings for participant P22

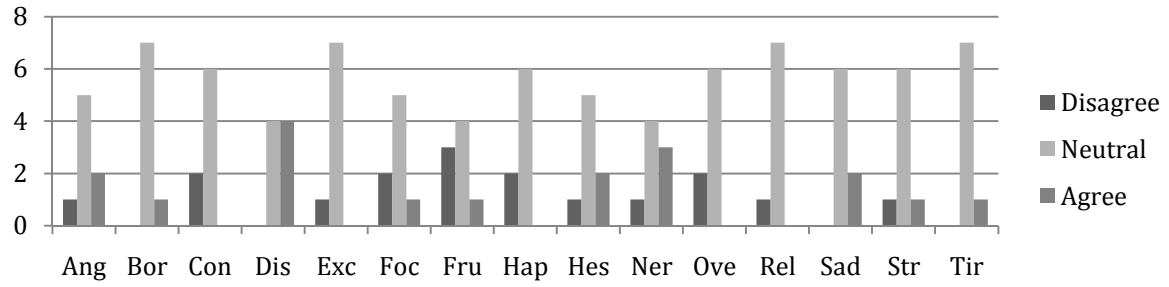


Emotional state ratings for participant P23



Emotional state ratings for participant P24




Emotional state ratings for participant P26
8
6
4
2
0
Ang
Bor
Con
Dis
Exc
Foc
Fru
Hap
Hes
Ner
Ove
Rel
Sad
Str
Tir
Disagree
Neutral
Agree

APPENDIX B PREDICTIVE MODEL RESULTS

Anger

Data set	Class distribution 1	2	3	Training Prediction Rate	Kappa	Test Prediction Rate	Kappa	Confusion matrix
Original	884	162	31	88.77	0.52	83.10	0.29	a b c <-- classified as 846 38 0 | a = 1 112 49 1 | b = 2 27 4 0 | c = 3
Under-sampled	31	31	31	82.80	0.74	26.88	-0.10	a b c <-- classified as 12 9 10 | a = 1 15 9 7 | b = 2 18 9 4 | c = 3
Resampled to 160 instances	160	160	160	91.25	0.87	51.67	0.28	a b c <-- classified as 66 55 39 | a = 1 45 84 31 | b = 2 36 26 98 | c = 3

Bored

Data set	Class distribution 1	2	3	Training Prediction Rate	Kappa	Test Prediction Rate	Kappa	Confusion matrix
Original	633	265	179	79.20	0.59	57.38	0.13	a b c <-- classified as 538 67 28 | a = 1 188 62 15 | b = 2 124 37 18 | c = 3
Under-sampled	179	179	179	73.00	0.60	38.36	0.08	a b c <-- classified as 63 55 61 | a = 1 42 73 64 | b = 2 60 49 70 | c = 3
Resampled to 160 instances	160	160	160	57.08	0.36	37.92	0.07	a b c <-- classified as 46 54 60 | a = 1 36 73 51 | b = 2 36 61 63 | c = 3

Confidence

Data set	Class distribution 1	2	3	Training Prediction Rate	Kappa	Test Prediction Rate	Kappa	Confusion matrix
Original	132	510	435	65.83	0.41	50.60	0.15	a b c <-- classified as 16 70 46 | a = 1 28 294 188 | b = 2 26 174 235 | c = 3
Under-sampled	132	132	132	72.98	0.59	37.88	0.07	a b c <-- classified as 58 29 45 | a = 1 46 44 42 | b = 2 55 29 48 | c = 3
Resampled to 160 instances	160	160	160	78.75	0.68	42.71	0.14	a b c <-- classified as 85 34 41 | a = 1 50 68 42 | b = 2 61 47 52 | c = 3

Distracted

Data set	Class distribution 1	2	3	Training Prediction Rate	Kappa	Test Prediction Rate	Kappa	Confusion matrix
Original	535	243	299	68.80	0.47	47.91	0.13	a b c <-- classified as 374 65 96 | a = 1 136 57 50 | b = 2 151 63 85 | c = 3
Under-sampled	243	243	243	76.95	0.65	42.52	0.14	a b c <-- classified as 109 65 69 | a = 1 65 85 93 | b = 2 61 66 116 | c = 3
Resampled to 160 instances	160	160	160	76.25	0.64	37.50	0.06	a b c <-- classified as 73 46 41 | a = 1 44 57 59 | b = 2 50 60 50 | c = 3

Excited

Data set	Class distribution			Training		Test		Confusion matrix
	1	2	3	Prediction Rate	Kappa	Prediction Rate	Kappa	
Original	465	489	123	81.62	0.68	54.60	0.22	a b c <-- classified as 297 148 20 | a = 1 190 270 29 | b = 2 56 46 21 | c = 3
Under-sampled	123	123	123	74.25	0.61	40.38	0.11	a b c <-- classified as 56 40 27 | a = 1 38 55 30 | b = 2 51 34 38 | c = 3
Resampled to 160 instances	160	160	160	81.46	0.72	44.38	0.17	a b c <-- classified as 78 48 34 | a = 1 53 64 43 | b = 2 50 39 71 | c = 3

Focused

Data set	Class distribution			Training		Test		Confusion matrix
	1	2	3	Prediction Rate	Kappa	Prediction Rate	Kappa	
Original	231	399	447	73.63	0.58	42.06	0.09	a b c <-- classified as 46 94 91 | a = 1 61 187 151 | b = 2 75 152 220 | c = 3
Under-sampled	231	231	231	67.82	0.52	37.81	0.07	a b c <-- classified as 106 60 65 | a = 1 93 81 57 | b = 2 92 64 75 | c = 3
Resampled to 160 instances	160	160	160	80.42	0.71	39.79	0.10	a b c <-- classified as 71 40 49 | a = 1 49 66 45 | b = 2 67 39 54 | c = 3

Frustration

Data set	Class distribution 1	2	3	Training Prediction Rate	Kappa	Test Prediction Rate	Kappa	Confusion matrix
Original	760	179	138	80.41	0.45	71.12	0.22	a b c <-- classified as 698 40 22 | a = 1 115 57 7 | b = 2 120 7 11 | c = 3
Under-sampled	138	138	138	67.87	0.52	44.93	0.17	a b c <-- classified as 67 26 45 | a = 1 38 73 27 | b = 2 65 27 46 | c = 3
Resampled to 160 instances	160	160	160	87.50	0.81	49.79	0.25	a b c <-- classified as 77 32 51 | a = 1 43 93 24 | b = 2 57 34 69 | c = 3

Happiness

Data set	Class distribution 1	2	3	Training Prediction Rate	Kappa	Test Prediction Rate	Kappa	Confusion matrix
Original	225	591	261	70.10	0.44	50.33	0.07	a b c <-- classified as 46 148 31 | a = 1 70 459 62 | b = 2 34 190 37 | c = 3
Under-sampled	225	225	225	77.48	0.66	45.19	0.18	a b c <-- classified as 106 49 70 | a = 1 59 93 73 | b = 2 65 54 106 | c = 3
Resampled to 160 instances	160	160	160	86.46	0.80	46.46	0.20	a b c <-- classified as 74 51 35 | a = 1 48 66 46 | b = 2 37 40 83 | c = 3

Hesitance

Data set	Class distribution			Training		Test		Confusion matrix
	1	2	3	Prediction Rate	Kappa	Prediction Rate	Kappa	
Original	710	284	83	80.59	0.53	68.25	0.25	a b c <-- classified as 624 72 14 | a = 1 172 108 4 | b = 2 70 10 3 | c = 3
Under-sampled	83	83	83	81.12	0.72	45.78	0.19	a b c <-- classified as 32 29 22 | a = 1 26 36 21 | b = 2 19 18 46 | c = 3
Resampled to 160 instances	160	160	160	79.17	0.69	54.17	0.31	a b c <-- classified as 82 44 34 | a = 1 61 79 20 | b = 2 34 27 99 | c = 3

Nervous

Data set	Class distribution			Training		Test		Confusion matrix
	1	2	3	Prediction Rate	Kappa	Prediction Rate	Kappa	
Original	841	182	54	78.09	0.00	75.49	0.10	a b c <-- classified as 788 46 7 | a = 1 151 21 10 | b = 2 47 3 4 | c = 3
Under-sampled	54	54	54	88.27	0.82	44.44	0.17	a b c <-- classified as 26 14 14 | a = 1 10 26 18 | b = 2 16 18 20 | c = 3
Resampled to 160 instances	160	160	160	87.29	0.81	51.46	0.27	a b c <-- classified as 83 42 35 | a = 1 42 61 57 | b = 2 24 33 103 | c = 3

Overwhelmed

Data set	Class distribution 1	2	3	Training Prediction Rate	Kappa	Test Prediction Rate	Kappa	Confusion matrix
Original	581	361	135	72.70	0.47	54.60	0.14	a b c <-- classified as 458 103 20 | a = 1 214 128 19 | b = 2 99 34 2 | c = 3
Under-sampled	135	135	135	86.91	0.80	36.30	0.04	a b c <-- classified as 51 38 46 | a = 1 41 44 50 | b = 2 53 30 52 | c = 3
Resampled to 160 instances	160	160	160	82.50	0.74	39.58	0.09	a b c <-- classified as 66 40 54 | a = 1 45 61 54 | b = 2 47 50 63 | c = 3

Relaxed

Data set	Class distribution 1	2	3	Training Prediction Rate	Kappa	Test Prediction Rate	Kappa	Confusion matrix
Original	310	528	239	74.56	0.56	49.12	0.14	a b c <-- classified as 134 140 36 | a = 1 93 377 58 | b = 2 54 167 18 | c = 3
Under-sampled	239	239	239	68.06	0.52	43.93	0.16	a b c <-- classified as 103 58 78 | a = 1 52 88 99 | b = 2 59 56 124 | c = 3
Resampled to 160 instances	160	160	160	79.58	0.69	40.63	0.11	a b c <-- classified as 84 26 50 | a = 1 40 52 68 | b = 2 55 46 59 | c = 3

Sad

Data set	Class distribution			Training		Test		Confusion matrix
	1	2	3	Prediction Rate	Kappa	Prediction Rate	Kappa	
Original	851	178	48	85.24	0.43	79.11	0.21	a b c <-- classified as 806 41 4 | a = 1 131 46 1 | b = 2 48 0 0 | c = 3
Under-sampled	48	48	48	84.03	0.76	39.58	0.09	a b c <-- classified as 20 11 17 | a = 1 13 19 16 | b = 2 13 17 18 | c = 3
Resampled to 160 instances	160	160	160	91.88	0.88	54.38	0.32	a b c <-- classified as 74 44 42 | a = 1 47 81 32 | b = 2 25 29 106 | c = 3

Stress

Data set	Class distribution			Training		Test		Confusion matrix
	1	2	3	Prediction Rate	Kappa	Prediction Rate	Kappa	
Original	694	267	116	81.71	0.58	64.07	0.19	a b c <-- classified as 600 71 23 | a = 1 168 86 13 | b = 2 91 21 4 | c = 3
Under-sampled	116	116	116	82.18	0.73	43.97	0.16	a b c <-- classified as 61 27 28 | a = 1 40 55 21 | b = 2 43 36 37 | c = 3
Resampled to 160 instances	160	160	160	82.71	0.74	42.08	0.13	a b c <-- classified as 86 35 39 | a = 1 53 59 48 | b = 2 57 46 57 | c = 3

Tired

Data set	Class distribution			Training		Test		Confusion matrix
	1	2	3	Prediction Rate	Kappa	Prediction Rate	Kappa	
Original	499	224	354	82.45	0.72	50.05	0.20	a b c <-- classified as 322 72 105 | a = 1 97 57 70 | b = 2 135 59 160 | c = 3
Under-sampled	224	224	224	83.78	0.76	39.88	0.10	a b c <-- classified as 96 64 64 | a = 1 65 81 78 | b = 2 65 68 91 | c = 3
Resampled to 160 instances	160	160	160	55.00	0.33	38.33	0.08	a b c <-- classified as 68 43 49 | a = 1 55 59 46 | b = 2 47 56 57 | c = 3

APPENDIX C SOFTWARE DESIGN DETAILS

The overall goal of the architecture (Figure C.1) was to provide a solution to the problem of recording and gathering users' subjective emotional state labels and user data for creating predictive models. The first step was to record subjective emotional state labels using a questionnaire. The second step was to record user data. We chose to record user data for the five minutes immediately before the questionnaire appeared. These data — subjective emotional state labels and user data — needed to be linked to each other so that the user data could be used to create models for the questionnaire responses received immediately after the user data was recorded. Regardless of the software solution, this process of recording user activity data and subjective emotional states is the same. These data also need to be accessible to researchers for predictive modeling.

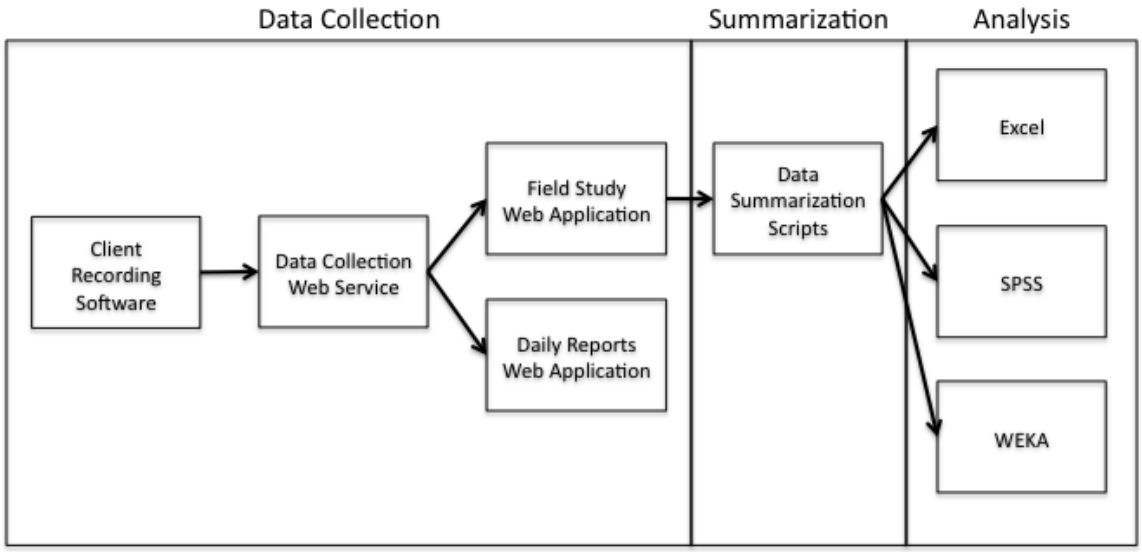


Figure C.1 - High-level architecture

Codifying the high-level process of recording and collecting subjective emotional state labels and user data is important because it solves many of the logistical issues of recording and

collecting data, determining ground truth, and making that data available for predictive modeling. Custom software for recording subjective emotional state labels and user data was necessary because although several software applications exist for collecting user data (e.g., AppMonitor [6], Windows Hooker [4]), there is no software system architecture that codifies the recording and collection process for both subjective user emotional states and user activity data. Furthermore, AppMonitor and Windows Hooker capture mouse and keyboard events, but a more general solution could capture other types of user activity data such as facial expression images that have already been shown to have high prediction rates [26].

In the rest of this appendix, we describe some similar software systems used for capturing user activity data and provide details of the four major components of our system: Client Recording Software, Data Collection Service, Data Retrieval Web Application, and Daily Reports Web Application.

C.1 Similar Software Systems

C.1.1 Physiological monitoring systems used in hospitals

Physiological monitoring systems used in hospitals seem to be more concerned with passing data between physiological sensing devices to a local measurement system and then on to a central recording system. Various papers have been examined in the areas of patient monitoring using wireless communication [35,54], wireless devices [62], network standards between bedside and central monitor stations [77], and body area networks [63,73,80]. The data model used by these systems would be useful to determine if their abstraction of the data is suitable for our use. Unfortunately, none of these provide an evaluation of the data model from different physiological measurement devices. Leaning et al. provide a high-level data model for intensive care systems [53], but fail to give an in-depth description of the monitoring data elements from physiological measurement devices such as EKGs.

ISO/IEEE 11073-10201 [5] describes a system for connecting medical devices to a healthcare management system using an "object-oriented systems management paradigm". It provides a

domain model. The standard consists of a collection of modules. The medical module describes some classes of interest. VMD is an abstraction of a medical device. Metric is a measurement. A Real Time Sample Array is used to represent "a real-time continuous waveform" such as a data from an electrocardiograph. A Time Sample Array represents a "non-continuous waveform" that is at fixed time intervals. The communication module also has some interesting classes related to communicating from the device to a bedside controller.

The domain model described in ISO/IEEE 11073-10201 is strongly typed and does not appear extensible. For example, the Metric class defines twenty-three attributes that seem to be intended to cover all possible measurable data types with a single numeric value. Perhaps the model covers data types from all the medical devices that existed at the time of publication, but it is not evident how the design can adapt to data from a device that does not fit the current model. Complex data types such as events that have more than one numeric value cannot be represented by this model. The complex type must be broken down into simpler data types with a single numeric value. This increases processing required on the device and should the complex data type be meaningful, it would have to be re-assembled eventually.

Generally, these systems are well-suited to monitoring and alerting health care professionals about the status of patients, but do not describe functionality for exporting their data for future analyses. For the affective computing researcher, the ability to perform analyses both during and after experiments is critical.

C.1.2 **SCADA systems**

I gained experience with a telemetry data capture system while working at a large natural gas transmission company. The system captured and visualized data that was received from a central SCADA system. The data points, called tags, originated from various locations and from various devices, and the data model needed to represent both locations and data types. Suction and discharge pressures, suction and discharge temperatures, calculated flows, valve states, gas compressibility and many other variables were captured in a generic fashion and visualizations were created with a custom graphing application.

Figure C.2 depicts a SCADA system. Data is gathered from disparate sources as tags and stored in a central location. These tags are stored in the central location for viewing by a visualization tool or transferred to other systems. The tags are an abstraction of the data recorded at the actual device. They also may represent calculated values such as compressibility constants or flow.

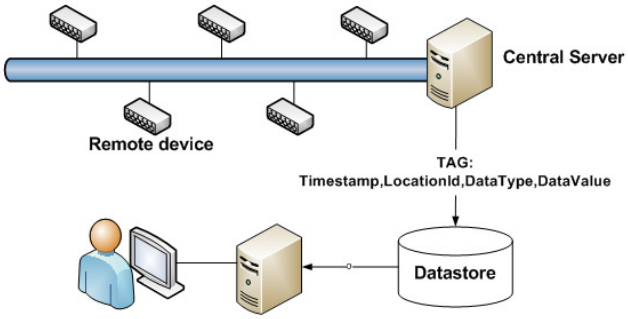


Figure C.2 - SCADA system architecture using tags to represent data points

Abstraction of data from affect measurement devices in a similar fashion could provide a plausible data model for capturing affect data from users. Further, this architecture may provide a good approach for collecting affect data from devices that may be added, removed, or swapped for newer ones.

One problem with this model, however, is that data is pushed from data sources at all times. In an affect data capture system, this may be undesirable. In fact, it may overwhelm the system as some data sources can generate a great amount of data. For example, a study may be interested in data from an electrocardiograph. The system may have the capability of measuring data from many sources such as a mouse, an electrocardiograph and an electroencephalograph. Mouse motion events can be generated within milliseconds of each other and in combination with data from other sources, i.e., the electrocardiograph and electroencephalograph, may cause the system to perform unreasonably slow.

C.1.3 **Oscilloscope Design**

Shaw and Garlan [72] describe the evolution of the design for an oscilloscope. Starting with a domain model using an object-oriented approach, the authors describe the transition to a layered

model, pipe-and-filter model, and finally a modified pipe-and-filter model. The domain model did not explain how varying types of data fit together, which component used what type of data. The layered model only allowed data to be passed between adjacent layers, thus preventing user access to data at lower layers that might be valuable. The pipe-and-filter model allowed great flexibility as each filter was regarded as a data transformer that could be chained together. However, the problem of users only being able to interact with the end-product of the pipeline still existed, as it did with the layered model. The modified pipe-and-filter model allowed users to interact and configure each filter through a control panel. This allowed users to modify the configuration of a filter that would affect the final output from the pipeline. Finally, specialized pipes were utilized to improve performance of expensive operations such as copying a large number of data structures.

The modified pipe-and-filter model allows great flexibility for transforming data and also configuring the transformation at steps along the way. This approach allows for dealing with complex data types that can be passed along the pipeline or transformed into simple numeric values. The oscilloscope design, however, does not address the problem of capturing data from multiple sources.

C.1.4 **Analysis Pattern - Observation and Measurements**

Fowler describes a number of object models for observing and measuring phenomena [28]. In the most relevant model, an Observation is made on a Person of some Phenomenon Type. The Observation consists of a Measurement or an Observation Category. A Measurement is of some Quantity. An Observation Category is of some Category. For example, the skin conductance level from the hand of a person is observed to be 1.5. This is a measurement of some quantity. An observation may also be made that a person is tall. This is a category observation of the height of an individual. This model allows for both quantitative and qualitative observations which are useful in our case because some observations may be that a mouse button is down.

C.1.5 **Relevance to our architecture**

The technical solution to our problem has similarities to the pipe-and-filter approach described by the oscilloscope design above. We have data sources (the questionnaire and mouse), we have filters (the data collection service and summarization scripts), and we have consumers (the predictive models). However, there are parts that are difficult to codify because the data source (e.g., mouse, keyboard, and facial expression images) may vary, thereby affecting the summarization scripts required. For this reason, we focus on codifying the recording and delivery mechanism in a way that is agnostic of the data source.

C.2 Client Recording Software

To collect data on participants' computers, software needed to be installed on participants' computers. The Client Recording Software was designed to record data on participants' computers and deliver it to a central location. If the features required for predictive modeling were known ahead of time, the software could generate those features and send them to researchers. This could result in a smaller amount of data sent to researchers, but not in all cases. Analysis of keystroke data in a study that used the same data we collected had many thousands of features because combinations of keystrokes -- digraphs for two-key combinations (e.g., t-h) and trigraphs for three keys (e.g., t-h-e) -- were used as features [25]. The resulting feature data was larger than the raw keystroke data collected. Selectively choosing to generate features for some data and not for others -- and sending those data to researchers -- may work in many cases. Raw mouse and keyboard events allowed us the most flexibility for generating features. Long after the study was over, we could generate new features if needed. We recommend collecting raw event data if possible to allow the most flexibility for feature extraction, but our architecture does not exclude sending data files that already have extracted features.

The rest of this section shows pseudo-code for the various components.

C.2.1 Core probe and questionnaire algorithms

Main() pseudocode:

```
# Kill any processes of this application that are still running
killAnyProcessesStillRunning()

# Verify all required configuration properties are set
verifyConfiguration()

started = false
while !started && tries++ < 10
    m_mainForm = new MainForm()
    running = true
    onError:
        log(error)

    if !started
        wait 1 minute
end while
```

MainForm() pseudocode:

```
# Restart timer restarts application every 12 hours, but only
# if questionnaire form is not showing.
startRestartTimer()

# Initialize the mouse and key logger
m_probeManager = new ProbeManager(settings)

# If there are files to send, send them
if m_probeManager.thereArePendingItemsToSend()
    m_probeManager.sendLogsToDataCollectionServer()
    onError:
        log(error)
```

MainForm.Load() pseudocode:

```
# If demographics haven't been collected, collect
m_demographicForm = new DemographicForm()
m_probeManager.sendLogsToDataCollectionServer()

# Start listening for mouse and keyboard events
m_probeManager.startListening()

# Start the questionnaire thread
startQuestionnaireThread()
```

Questionnaire thread pseudocode:

```
while running
    # If the questionnaire interval (1 hour) has elapsed
    if questionnaireIntervalHasElapsed
        # If user is not active, sleep for 2 minutes and check again
        if !m_probeManager.userIsActive()
            sleep 2 minutes
            continue

        # If the user actively skipped the questionnaire, don't show it for
        # another 30 minutes
        if timeSinceLastSkip < 30 minutes
            log("user is ignoring")
            sleep until 30 minutes has elapsed since timeSinceLastSkip
            continue

        # If the questionnaire is already appearing, move along
        if questionnaire is visible
            continue

        show taskbar balloon     # requests user to fill out questionnaire
    end if
end while
```

ProbeManager.Start() pseudocode:

```
initialize m_logEventQueue
initialize m_broadcasterEventQueue
start broadcaster thread         # broadcasts events to listeners
start mouse event hook           # fires on every mouse event
start keyboard event hook        # fires on every keyboard event
start window process timer       # fires every 10 minutes
```

Mouse event hook fired pseudocode:

```
e = new LoggableMouseEvent from mouse event
m_broadcasterEventQueue.add(e)
```

Keyboard event hook fired pseudocode:

```
e = new LoggableKeyboardEvent from keyboard event
m_broadcasterEventQueue.add(e)
```

Broadcaster thread pseudocode:

```
while running
    n = m_broadcasterEventQueue.count
    for i = 1 to n               # only process elements in queue when loop started
        e = m_broadcasterEventQueue.pop()
        lock m_broadcasterEventQueue
            if e is mouse or keyboard event
                m_activityMonitor.userEventOccurred(e.timestamp)
            m_loggerEventQueue.add(e)
        end lock
        i++
    end for
end while
```

Questionnaire completed pseudocode:

```
qe = new QuestionnaireEvent()
m_mainForm.questionnaireComplete(qe)
```

MainForm.questionnaireComplete() pseudocode:

```
e = ExtractSystemInformation()
m_probeManager.loggerEventQueue.add(e)
m_probeManager.loggerEventQueue.add(questionnaireEvent)
m_probeManager.loggerEventQueue.writePendingLogs()
m_probeManager.sendLogsToDataCollectionServer()
```

C.2.2 SEND ALGORITHMS

ProbeManager.sendLogsToDataCollectionServer() pseudocode:

```
DirectorySender.SendExperimentFiles(experimentName, participantId, logDirectory)
```

C.3 Data Collection Service

We recognized that the problem of collecting data files from participant computers was a more general problem, so we designed and implemented a general solution consisting of a client API (the Data Collection Service API) and server-side service (Data Collection Web Service). The client API was responsible for reliably sending data to the server-side service. The server-side service was responsible for receiving and storing participant data. The benefit of a general solution is that it allows any type of data (text or binary) and it is re-usable for other experiments. A drawback of a general solution is that metadata such as the type of data can be difficult to apply to the data. In our case, we encoded metadata in filenames which allowed us to indicate whether the file contained questionnaire, mouse motion, or keystroke data. The remainder of this section describes the Data Collection Service API and the Data Collection Web Service.

C.3.1 Data Collection Service API

DirectorySender.SendExperimentFiles() pseudocode:

```
sendableFiles = getSendableFiles()
moveToSendingFolder(sendableFiles)
filesInSending = getFilesInSending()
foreach file in filesInSending
    shortFileName = getShortFileName(file)
    base64EncodedContents = GetContentsAsBase64(file)
    webService = getWebService()
    webService.uploadFile(experimentName, participantId, shortFileName,
base64EncodedContents)
    onError:
        log(error)
        break
    moveToSent(file)
end foreach
```

C.3.2 Data Collection Web Service

WebService.uploadFile() pseudocode:

```
decodedFileContents = base64Decode(base64EncodedContents)
folder = EXPERIMENT_DATA_ROOT+"/"+experimentName+"/"+participantId
folder.mkdirs()
fullname = folder+"/"+shortFileName
writeToFile(fullname, decodedFileContents)
```

C.4 Data Retrieval Web Application

The data retrieval web application was an essential part of the data collection service for viewing and retrieving participant data after it has been sent to the data collection service. This web application utilized the Google Web Toolkit with the JavaScript EXT extension, more commonly known as GWT-EXT. See Chapter 3 for more details.

C.5 Daily Reports Web Application

The daily reports web application provided daily information via email (emails were also archived and viewable using a browser) specifically about the field study we conducted. It provided information about participants' progress in the study including whether the demographic questionnaire had been completed and the number of ESQs completed. See Chapter 3 for more details.